

Hinweise zum Upgrade von LabVIEW™

Version 8.0

In dieser Broschüre wird beschrieben, wie LabVIEW unter Windows, Mac OS und Linux auf Version 8.0 aktualisiert wird, welche Probleme beim Upgrade auftreten können und um welche Funktionen die neue LabVIEW-Version erweitert wurde.

Weitere Informationen zu neuen Funktionen in LabVIEW 8.0 und zu Methoden der LabVIEW-Programmierung, Schritt-für-Schritt-Anleitungen sowie Hinweise zu den LabVIEW-VIs, -Funktionen, -Paletten, -Menüs und -Werkzeugen finden Sie in der *LabVIEW-Hilfe*. In der *LabVIEW-Hilfe* wird auch auf weitere Dokumentation zu LabVIEW von National Instruments verwiesen. Zum Öffnen der *LabVIEW-Hilfe* klicken Sie auf **Hilfe» LabVIEW-Hilfe durchsuchen**.

Inhaltsverzeichnis

Upgrade auf LabVIEW 8.0	3
Konvertieren von VIs.....	3
Upgrade von Toolkits, Instrumententreibern und Erweiterungspaketen (Add-Ons)	4
Upgrade zusätzlicher Software von National Instruments	4
Upgrade älterer LabVIEW-Versionen	5
Hinweise zur Kompatibilität beim Upgrade	6
Upgrade von LabVIEW 7.x	6
Upgrade von LabVIEW 6.x	26
Upgrade von LabVIEW 5.x	32
Upgrade von LabVIEW 4.x	33
Upgrade von LabVIEW 3.x oder älteren Versionen.....	35
Änderungen und Neuerungen in LabVIEW 8.0.....	35
Aktivieren der LabVIEW-Lizenz	36
Starten von LabVIEW	36
An- und Abmelden in LabVIEW.....	36
Domain-Kontomanager	36
Neuerungen am Dialogfeld “Neu”	37

RT-, FPGA- und PDA-Systeme	37
LabVIEW-Projekte.....	38
LabVIEW-Projektbibliotheken.....	46
Übertragung aktueller Frontpanel-Werte mit Umgebungsvariablen	47
Erstellen von Quellcodepaketen	50
Neuerungen am Application Builder.....	51
Neue Palettenaufteilung.....	53
Neuordnung des Menüs.....	57
Verwenden der Versionsverwaltung	58
Verbesserungen an VIs und Funktionen.....	60
Neuerungen beim VI-Hierarchie-Fenster	77
Verbesserungen am Frontpanel	77
LabVIEW-MathScript (Windows)	91
Matrix-Datentyp	91
Verbessertes Dialogfeld “Optionen”	94
Dialogfelder “Speichern unter” und “Für vorige Version speichern”	95
Verbesserungen bezüglich der Dateigröße.....	95
Verwendung von DLLs in mehreren LabVIEW-Versionen	95
Festlegen der Fensterposition bei Ausführung.....	95
Einstellen von Sprachen des Laufzeitmenüs	96
Verbesserungen am VI-Server	96
Klassen-Browser-Fenster	98
Verwenden von .NET-Assemblies	98
Suchmaschine für Gerätetreiber	99
Assistent für Gerätetreiber-VIs.....	100
Assistent für Gerätetreiberprojekte.....	100
Verwenden von ablaufinvarianten VIs und Fehlersuche.....	101
Erstellen von Sonden nach Start eines VIs.....	102
Profil-VIs in mehreren Zielsystemen	102
64-Bit-Integer-Datentyp	103
NI Spy.....	103
Suchen und Ersetzen.....	104
Individuelle Tastenkombinationen	104
Neuerungen im Dialogfeld “Drucken”	104
Bearbeiten von VI-Symbolen	105
Neuerungen und Änderungen an der Dokumentation.....	105
Neuerungen an der Beispielsuchmaschine	109
Andere Änderungen und Neuerungen in LabVIEW 8.0	110

Upgrade auf LabVIEW 8.0

Wenn Sie Ihre bestehende LabVIEW-Version auf 8.0 aktualisieren möchten, lesen Sie zunächst die Abschnitte *Konvertieren von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW x.x*, wobei *x.x* für Ihre LabVIEW-Version steht.

Konvertieren von VIs

VIs, die zuletzt in einer LabVIEW-Version ab 4.0 gespeichert wurden, werden beim Öffnen in LabVIEW 8.0 automatisch konvertiert und kompiliert. Um zu verhindern, dass das VI bei jedem Öffnen umgewandelt und neu kompiliert wird – was zusätzlichen Arbeitsspeicher kostet – sollte das VI am besten in LabVIEW 8.0 gespeichert werden.

VIs, an denen ungespeicherte Änderungen vorgenommen wurden – dazu zählt unter anderem auch eine Neukompilierung – können sich merklich verlangsamen. Weitere Informationen zu diesem Thema und zum Speichern von VIs finden Sie in der *LabVIEW-Hilfe*.



Hinweis In LabVIEW 8.0 gespeicherte VIs lassen sich nicht in älteren Versionen öffnen. Um VIs in LabVIEW 7.1 ausführen zu können, wählen Sie im Menü **Datei»Für vorige Version speichern**. Es empfiehlt sich, eine Sicherheitskopie aller VIs anzulegen, die auch in älteren LabVIEW-Versionen genutzt werden sollen.

Wenn Ihr Rechner nicht genug Speicherplatz hat, um alle VIs auf einmal zu konvertieren, gehen Sie bei der Umwandlung etappenweise vor. Werfen Sie unter Umständen auch einen Blick auf die Hierarchie der VIs, die konvertiert werden sollen. Laden und speichern Sie zunächst die SubVIs der unteren Hierarchiestufen. Fahren Sie in der Hierarchie aufwärts fort. Das Haupt-VI sollte zum Schluss geöffnet und konvertiert werden. Um VIs verzeichnisweise umzuwandeln, wählen Sie **Werkzeuge»Fortgeschritten»Massenkompilierung**. VIs eines Verzeichnisses oder einer LLB werden in alphabetischer Reihenfolge kompiliert. Wird während des Umwandlungsprozesses zuerst ein Haupt-VI gefunden, wird für die Massenkompilierung in etwa genau so viel Arbeitsspeicher benötigt, als ob dieses VI zuerst geöffnet worden wäre.

Die Speicherauslastung können Sie sich unter **Hilfe»Über LabVIEW** anzeigen lassen.

Upgrade von Toolkits, Instrumententreibern und Erweiterungspaketen (Add-Ons)

Installieren Sie nach Abschluss der Installation von LabVIEW 8.0 alle Toolkits und Erweiterungspakete für das Programm in das Programmverzeichnis von LabVIEW 8.0. Eventuell müssen Sie zunächst das Toolkit früherer LabVIEW-Versionen deinstallieren. Weitere Informationen zur Installation finden Sie in der Dokumentation des LabVIEW-Toolkits oder Add-Ons.

Toolkits, Instrumententreiber und Zusatz-VIs aus älteren Versionen müssen für LabVIEW 8.0 ebenfalls massenkompiliert werden. Weitere Informationen zur Massenkompilierung von VIs finden Sie im Abschnitt *Konvertieren von VIs* in diesem Dokument.

Folgende Toolkits, Hardware-Treiber und Erweiterungspakete müssen für LabVIEW 8.0 entweder aktualisiert oder heruntergeladen werden:

- Für den LabVIEW Application Builder ist ein separates Upgrade auf den LabVIEW Application Builder 8.0 erforderlich. Im LabVIEW 8.0 Professional Development System ist der Application Builder 8.0 bereits enthalten. Weitere Informationen zur Installation des LabVIEW Application Builders finden Sie in der *LabVIEW Application Builder Readme* im Verzeichnis `labview\readme`.
- Für LabVIEW 8.0 wird der VI-Analyzer 1.1 benötigt. Dieses Programm, zusammen mit Hinweisen zum Upgrade, finden Sie auf unserer Website ni.com/info nach Eingabe des Infocodes `exd8yy`.
- Für die Verwendung des Internet Toolkits 6.0 mit LabVIEW 8.0 müssen zusätzliche VIs heruntergeladen werden. Zum Herunterladen der VIs besuchen Sie bitte unsere Website ni.com/info und geben Sie den Infocode `exkrkb` ein.
- Der Treiber für das digitale Multimeter 34401A von HP (jetzt: Agilent) ist der Treibervorlage für Digitalmultimeter von National Instruments ähnlicher geworden. Dieser Treiber ist daher nicht mit dem Treiber HP34401A kompatibel, der in früheren LabVIEW-Versionen enthalten war. Wenn Sie den Treiber von LabVIEW 7.x benötigen, laden Sie ihn von der Website ni.com/idnet herunter.

Upgrade zusätzlicher Software von National Instruments

Für LabVIEW 8.0 wird eine NI-TestStand-Version ab 3.5 benötigt. Dieses Programm, zusammen mit Hinweisen zum Upgrade, finden Sie auf der Website ni.com/info, wenn Sie den Infocode `exd8yy` eingeben.

Für LabVIEW 8.0 wird eine Logos-Version ab 4.6 benötigt. Zum Herunterladen der neuesten Version von Logos besuchen Sie die Website ni.com/info und geben Sie den Infocode `exymjt` ein.

Upgrade älterer LabVIEW-Versionen

Ein Upgrade von LabVIEW hat keine Auswirkungen auf die vorhandene Version, da die neue Version in ein anderes Verzeichnis installiert wird. So lautet das Programmverzeichnis von LabVIEW-Versionen bis 5.x beispielsweise `labview`. Alle LabVIEW-Versionen ab 6.0 werden im Verzeichnis `labview x.x` gespeichert (`x.x` steht für die Version). Sie können LabVIEW 8.0 ohne vorherige Deinstallation älterer Versionen installieren.

Sie können die vorhandene Version jedoch auch deinstallieren, die Installation von LabVIEW 8.0 starten und das Installationsverzeichnis auf dasselbe `labview`-Verzeichnis wie bei der letzten Version einstellen.

(Windows) Die bisherige LabVIEW-Version kann auch in der Systemsteuerung unter "Software" deinstalliert und durch LabVIEW 8.0 ersetzt werden. Dateien, die im `labview`-Verzeichnis erstellt wurden, bleiben vom Deinstallationsprogramm unberührt.



Hinweis Bei der Deinstallation oder Neuinstallation von LabVIEW werden alle `*.lib`-Dateien im Verzeichnis `vi.lib` einschließlich der darin befindlichen VIs und Elemente entfernt. Speichern Sie Ihre VIs und Elemente daher im Verzeichnis `user.lib`. So werden die VIs auch den Paletten **Funktionen** und **Elemente** hinzugefügt.

Um die Einstellungen zur LabVIEW-Oberfläche von der letzten Version zu übernehmen, kopieren Sie die entsprechende Datei aus dem `labview`-Verzeichnis.



Vorsicht! Alle bis dato vorgenommenen Änderungen an den Grundeinstellungen werden damit in der neuen Version überschrieben.

Kopieren Sie die Datei mit den Einstellungen nach der Installation in das Programmverzeichnis von LabVIEW 8.0.

(Windows) Die Einstellungen sind in der Datei `labview.ini` gespeichert.

(Mac OS) Die Datei mit den Einstellungen lautet `LabVIEW Preferences` und befindet sich im Stammverzeichnis unter `Library:Preferences`.

(Linux) LabVIEW speichert die Einstellungen unter dem Namen `.labviewrc` im Stammverzeichnis.

Um Dateien aus dem Verzeichnis `user.lib` von der bisherigen Version zu übernehmen, kopieren Sie die Dateien mit den Einstellungen in das gewünschte `labview`-Verzeichnis. Kopieren Sie die Dateien nach der Installation von LabVIEW 8.0 in das Verzeichnis `user.lib` des neuen LabVIEW-Verzeichnisses.

Hinweise zur Kompatibilität beim Upgrade

In den nachfolgenden Abschnitten werden Probleme beim Upgrade und bezüglich der Kompatibilität verschiedener LabVIEW-Versionen besprochen.

Upgrade von LabVIEW 7.x

Bei einem Upgrade von LabVIEW 7.x auf LabVIEW 8.0 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW 8.0 folgende Änderungen:

- LabVIEW 7.1 und spätere Versionen laufen nicht unter Windows Me/98/95. LabVIEW 8.0 läuft nicht unter Windows NT.
- Zwar läuft LabVIEW 8.0 auch unter Windows XP x64, aber aktuelle Treiber arbeiten nur bedingt unter Windows XP x64.
- LabVIEW 8.0 läuft weder unter Mac OS X 10.2 noch unter Vorgängerversionen.
- LabVIEW 8.0 läuft nicht unter Sun Solaris.

Systemanforderungen

Für LabVIEW 7.x wird ein Arbeitsspeicher von mindestens 128 MB benötigt. Empfohlen werden jedoch 256 MB. Für LabVIEW 8.0 ist ein Arbeitsspeicher von mindestens 256 MB erforderlich. Empfohlen werden jedoch 512 MB.

Für LabVIEW 7.x ist eine Bildschirmauflösung von 800×600 Pixeln erforderlich. Empfohlen werden jedoch 1.024×768 Pixel. Für LabVIEW 8.0 ist eine Bildschirmauflösung von 1.024×768 Pixeln erforderlich.

(Windows) LabVIEW 7.x arbeitet nur mit einem Pentium III, einem Celeron-Prozessor mit 600 MHz oder einem gleichwertigen Produkt. Es wird jedoch ein Pentium 4 oder ein gleichwertiger Prozessor empfohlen. Für LabVIEW 8.0 ist ein Pentium III bzw. ein Celeron-Prozessor mit 866 MHz oder ein gleichwertiges Produkt Mindestanforderung. Es wird jedoch ein Pentium 4/M oder ein gleichwertiger Prozessor empfohlen.

(Linux) LabVIEW 7.x arbeitet nur mit einem Pentium III, einem Celeron-Prozessor mit 600 MHz oder einem gleichwertigen Produkt. Es wird jedoch ein Pentium 4 oder ein gleichwertiger Prozessor empfohlen. Für LabVIEW 8.0 ist ein Pentium III bzw. ein Celeron-Prozessor mit 866 MHz

oder ein gleichwertiges Produkt Mindestanforderung. Es wird jedoch ein Pentium 4/M oder ein gleichwertiger Prozessor empfohlen.

(Windows) Für LabVIEW 7.x ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 130 MB bzw. für die Vollversion ein Speicherplatz von 550 MB erforderlich. Für LabVIEW 8.0 ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 900 MB bzw. für die Vollversion ein Speicherplatz von 1,2 GB erforderlich.

(Mac OS) Für LabVIEW 7.x ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 280 MB bzw. für die Vollversion ein Speicherplatz von 350 MB erforderlich. Für LabVIEW 8.0 ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 500 MB bzw. für die Vollversion ein Speicherplatz von 600 MB erforderlich.

(Linux) Für LabVIEW 7.x ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 200 MB bzw. für die Vollversion ein Speicherplatz von 300 MB erforderlich. Für LabVIEW 8.0 ist für die LabVIEW-Minimalversion ein Festplattenspeicherplatz von 400 MB bzw. für die Vollversion ein Speicherplatz von 550 MB erforderlich.

(Linux) Für LabVIEW 7.x wird die GNU C Library (`glibc`) ab Version 2.1.3 benötigt. Empfohlen wird eine Version ab 2.2.4. Für LabVIEW 8.0 ist die GNU C Library ab Version 2.2.4 erforderlich.

(Linux) LabVIEW 7.x läuft nur unter Red Hat Linux ab 7.0, Mandrake Linux ab 8.0, SuSE Linux ab 7.1 oder Debian Linux ab 3.0. LabVIEW 8.0 arbeitet nur mit Red Hat Enterprise Linux WS ab 3, MandrakeLinux/Mandriva ab 10.0 oder SuSE Linux ab 9.1.

Benutzerdefinierte Palettenansichten

In LabVIEW 8.0 gibt es keine benutzerdefinierten Palettenansichten mehr. Paletten können jetzt ohne Anlegen einer benutzerdefinierten Palettenansicht bearbeitet werden. Alle Änderungen an den Paletten in LabVIEW 8.0 sind im Abschnitt *Verbesserungen bei der Palettenbearbeitung* beschrieben.

Änderungen im Verhalten von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW 8.0 geändert.

.NET-VIs und -Anwendungen

Für die .NET-Funktionen und -Anwendungen in LabVIEW 8.0 wird mindestens das .NET Framework 1.1 Service Pack 1 benötigt. Vor der Installation des .NET Framework 1.1 Service Pack 1 muss Microsoft .NET Framework 1.1 Hotfix KB886904 entfernt werden.

Wenn Sie ein .NET-VI laden, das zuletzt in LabVIEW 7.x gespeichert wurde, werden Sie möglicherweise dazu aufgefordert, die Assemblys zu suchen, mit denen das VI arbeitet. Das gilt auch dann, wenn sich die Assembly-Dateien im selben Verzeichnis wie das VI befinden oder wenn sie im Dialogfeld **Werkzeuge»Fortgeschritten»NET-Assembly-Referenzen** in LabVIEW 7. x registriert wurden.

Algorithmen der Analyse-VIs

In LabVIEW-Versionen ab 7.1 arbeiten Analyse-VIs mit dem BLAS/LAPACK-Algorithmus. Dadurch liefern diese VIs nun genauere Ergebnisse. In LabVIEW 8.0 befinden sich diese VIs auf den Paletten **Mathematik** und **Signalverarbeitung**.

Express-VI “Signale anfügen”

Wenn in LabVIEW 7.x beim Express-VI “Signal anfügen” der Eingang **Eingangssignal A** ein leeres Array ist oder offen gelassen wird und Sie den Eingang **Eingangssignal B** mit einem einzelnen oder kombinierten Signal verbinden, ist auch der Ausgang **Angefügte Signale** leer. Wenn es in LabVIEW 8.0 kein **Eingangssignal A** gibt oder dieser Eingang offen gelassen wird und Sie an **Eingangssignal B** ein einzelnes Signal anlegen, gibt das Express-VI **Eingangssignal B** aus. Wenn **Eingangssignal B** ein kombiniertes Signal enthält, werden alle Signale, aus denen es sich zusammensetzt, aneinander angehängt.

Vergleichsfunktionen

In LabVIEW 7.x und Vorgängerversionen werden Variantdaten von den Vergleichsfunktionen zuerst in Bezug auf die Länge und anschließend bitweise verglichen. In LabVIEW 8.0 beginnt der Vergleich der Variantdaten mit den Typencodes, in denen die tatsächliche Typinformation der Variantdaten kodiert ist, und wird dann bei den anderen typspezifischen Attributen fortgeführt.

VI “Skalarprodukt”

In LabVIEW 7.0 wird das Skalarprodukt der Eingangsvektoren X und Y beim VI “Skalarprodukt” mit Hilfe folgender Gleichung berechnet:

$$X^*Y = \sum_{i=0}^{n-1} x_i y_i$$

In LabVIEW-Versionen ab 7.1 wird das Skalarprodukt komplexer Zahlen dagegen mit folgender Gleichung berechnet:

$$X^*Y = \sum_{i=0}^{n-1} x_i y_i^*$$

wobei y_i^* die komplex Konjugierte von y_i ist.

VI “Einfacher Textreport” (Mac OS und Linux)

Das Anschlussfeld des VIs “Einfacher Textreport” hat sich geändert. Wenn Sie in LabVIEW 8.0 ein VI öffnen, in dem das VI “Einfacher Textreport” verwendet wird und das zuvor in LabVIEW 7.x oder einer älteren Version gespeichert wurde, müssen Sie das SubVI mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **SubVI neu zuweisen** auswählen.

Funktion “In String formatieren”

Wenn Sie in LabVIEW 7.x die %- und %x-Syntaxelemente für Formatbezeichner mit der Funktion “In String formatieren” verwenden, werden die Werte des Fließkommazahl-Eingangs auf 32-Bit-Integer gerundet, bevor die Eingänge in einen String konvertiert werden.

In LabVIEW 8.0 rundet diese Funktion aufgrund der Syntaxelemente für Formatbezeichner Fließkomma-Eingänge auf 64-Bit-Integer, bevor die Eingangswerte in Strings konvertiert werden.

Funktion “Zahlen verbinden”

In LabVIEW 7.x und Vorgängerversionen konvertiert die Funktion “Zahlen verbinden” 32-Bit-Integer-Eingänge in 16-Bit-Integer, um einen 32-Bit-Integer zu erstellen. In LabVIEW 8.0 wird mit der Funktion “Zahlen verbinden” aus 32-Bit-Integer-Eingängen ein 64-Bit-Integer erzeugt.



Hinweis Wenn Sie ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.0 öffnen, wandelt LabVIEW 32-Bit-Integer-Eingänge in 16-Bit-Integer um.

Funktionen für die Konvertierung von Zahlen in Strings

In LabVIEW 7.x werden Fließkommazahlen bei den Funktionen “Zahl nach Hexadezimal-String”, “Zahl nach Oktal-String” und “Zahl in Dezimal-String” vor der Umwandlung in einen String auf einen 32-Bit-Integer gerundet.

In LabVIEW 8.0 runden diese Funktionen Fließkommazahlen auf 64-Bit-Integer, bevor die Eingangswerte in Strings konvertiert werden. Wenn Sie jedoch ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.0 öffnen, rundet LabVIEW aus Gründen der Kompatibilität und Funktion die Fließkommazahlen auf 32-Bit-Integer.

Funktion “VI-Referenz öffnen”

Wenn in LabVIEW 7.x der Eingang **VI-Pfad** der Funktion “VI-Referenz öffnen” ein Pfad ist und sich im Speicher ein VI mit dem gleichen Namen befindet, gibt LabVIEW eine Referenz auf das VI im Speicher aus, selbst wenn der Pfad zu dem VI im Speicher nicht dem angegebenen Pfad entspricht.

Ist der Eingang **VI-Pfad** von “VI-Referenz öffnen” ein String, wird ein VI nur geöffnet, wenn **VI-Pfad** mit dem vollständigen Namen des VIs im Speicher des Zielsystems übereinstimmt. Wenn **VI-Pfad** ein Pfad ist, sucht LabVIEW nach einem VI im Speicher mit demselben Zielpfad. Wenn kein VI mit passendem Pfad gefunden wird, versucht LabVIEW, das VI unter dem angegebenen Pfad zu laden. Bei Fehlen des VIs oder bei Konflikten mit anderen VIs im Speicher wird eine Fehlermeldung ausgegeben. Weitere Informationen zur richtigen Schreibweise von Dateinamen finden Sie im Abschnitt [LabVIEW-Projektbibliotheken](#) dieses Dokuments. Weitere Informationen zu Zielsystemen finden Sie im Abschnitt [LabVIEW-Projekte](#) dieses Dokuments.

VI “Schnellskalierung”

Wenn in LabVIEW 7.1 und Vorgängerversionen der Eingang **X** des VIs “Schnellskalierung 1D” bzw. “Schnellskalierung 2D” ein Array aus Nullen ist, gibt das VI **max|X|** als **0** und **Y[i]=X[i]/Max|X|** bzw. **Yij=Xij/Max|X|** als Array aus NaN aus. Wenn in LabVIEW 8.0 der Eingang **X** des VIs ein Array aus Nullen ist, gibt das VI unter **max|X|** den Wert **0** und unter **Y[i]=X[i]/Max|X|** bzw. **Yij=Xij/Max|X|** als Array aus Nullen aus.

VI “Schlüssel lesen”

In LabVIEW 7.x und Vorgängerversionen können mit Hilfe des VIs “Schlüssel lesen” japanische Multibyte-Zeichen gelesen werden, die in Shift-JIS kodiert sind. Dazu muss der Eingang **Multibyte-Codierung** mit 1 oder <Shift-JIS> verbunden werden. In LabVIEW 8.0 liest das VI “Schlüssel lesen” kodierte Multibyte-Zeichen automatisch. Dazu muss nur die Ländereinstellung des Betriebssystems auf die entsprechende Codierung gesetzt werden.

VI “Skalieren”

Wenn in LabVIEW 7.1 und Vorgängerversionen der Eingang **X** des VIs “1D skalieren” bzw. “2D skalieren” ein Array aus Nullen ist, gibt das VI unter **Skalierung** den Wert **0**, unter **Offset** den Wert **0** und unter **Y=(X-Offset)/Skalierung** ein Array aus NaN aus. Wenn in LabVIEW 8.0 und Vorgängerversionen der Eingang **X** des VIs “Skalieren” ein Array aus Nullen ist, gibt das VI unter **Skalierung** den Wert **1**, unter **Offset** den Wert **0** und unter **Y=(X-Offset)/Skalierung** ein Array aus Nullen aus.

Semaphor-VIs

In LabVIEW 7.x wird nicht versucht, die VIs “Semaphor freigeben” und “Semaphor belegen” auszuführen, wenn der Parameter **Fehler (Eingang)** einen Fehler enthält. In LabVIEW 8.0 werden diese VIs trotzdem ausgeführt, selbst wenn der Parameter **Fehler (Eingang)** einen Fehler enthält. Wenn Sie allerdings ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.0 öffnen, wird die Vorgehensweise von LabVIEW 7.x beibehalten.

SMTP-E-Mail-VIs

In LabVIEW 7.x und Vorgängerversionen können Sie einen Zeichensatz festlegen, indem Sie einen Wert mit dem Eingang **Zeichensatz** eines SMTP-E-Mail-VIs verbinden. In LabVIEW 8.0 wird bei den SMTP-E-Mail-VIs davon ausgegangen, dass die Nachricht im Zeichensatz des Systems vorliegt. Mit diesen VIs wird die Nachricht vor dem Versenden im UTF-8-Format kodiert. Die Eingänge **Zeichensatz** und **Zeichenkonvertierung** gibt es bei den neuen SMTP-E-Mail-VIs nicht mehr.

VI “Komplexe Zahlen sortieren”

Wenn Sie in LabVIEW 7.x und Vorgängerversionen den Eingang **Methode** des VIs “Komplexe Zahlen sortieren” auf **Betrag** stellen, wird die Reihenfolge der Elemente mit dem gleichen Betrag nicht geändert. Wenn Sie **Methode** in LabVIEW 8.0 auf **Betrag** setzen, werden die Elemente mit gleichem Betrag zuerst nach ihrem Real- und dann nach ihrem Imaginärteil sortiert.

VI “Einheitsvektor”

In LabVIEW 7.x und Vorgängerversionen wird die Norm des Eingangsvektors beim VI “Einheitsvektor” anhand der folgenden Gleichung berechnet:

$$\|X\| = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

In LabVIEW 8.0 berechnet das VI “Einheitsvektor” die Norm eines Eingangsvektors dagegen mittels folgender Gleichung:

$$\|X\| = \left(|x_0|^y + |x_1|^y + \dots + |x_{n-1}|^y \right)^{\frac{1}{y}}$$

wobei X der Vektor (Eingang), $\|X\|$ die Norm und y der Normtyp ist.

Benutzer-VIs

VIs, die Sie in den Verzeichnissen `labview\help`, `labview\project` und `labview\wizard` ablegen, werden jeweils in den Menüs **Hilfe**, **Werkzeuge** und **Datei** angezeigt. VIs, die Sie in LabVIEW 7.x und Vorgängerversionen in diesen Verzeichnissen ablegen, funktionieren möglicherweise nicht wie erwartet in LabVIEW 8.0, da diese VIs in LabVIEW 8.0 in einer privaten Kopie der Anwendung geöffnet werden.

Eine Liste aller Benutzer-VIs im Arbeitsspeicher aller Applikationsinstanzen erhalten Sie mit Hilfe des VIs “VIMemory Get VIs in Memory” in der Bibliothek `labview\vi.lib\Utility\allVIsInMemory.llb`. Eine Referenz für die aktuelle Applikationsinstanz wird mit dem VI “Get User Application Reference” in der Bibliothek `labview\vi.lib\Utility\allVIsInMemory.llb` erzeugt. Alles Weitere zu Applikationsinstanzen ist im Abschnitt [Arbeiten mit mehreren Kopien einer Anwendung](#) beschrieben.

Veraltete VIs und Funktionen

Folgende VIs und Funktionen werden von LabVIEW 8.0 nicht unterstützt:

- Die PPC-VIs werden bei LabVIEW-Versionen ab 7.1 nicht mehr installiert. Stattdessen gibt es jetzt die TCP-VIs.
- Das VI “QR-Faktorisierung” wird von LabVIEW 8.0 nicht mehr unterstützt. Verwenden Sie stattdessen das VI “QR-Zerlegung”.
- Die VIs “Levenberg Marquardt” und “Nichtlineare Levenberg-Marquardt-Anpassung” sind in LabVIEW 8.0 durch das VI “Nichtlineare Kurvenanpassung” ersetzt worden.

- In LabVIEW 8.0 ist die Funktion “VISA: Statusinformation” nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie anstelle dieser Funktion das VI “Einfacher Fehlerbehandler” oder “Allgemeiner Fehlerbehandler”.
- Die VIs “Chi-Quadrat-Verteilung”, “F-Verteilung”, “Normalverteilung” und “T-Verteilung” wurden in LabVIEW 8.0 durch die Instanzen “Chi-Quadrat”, “F”, “Normal” und “Student-t” des VIs “Stetige Verteilung” ersetzt.
- Die VIs “Inverse Chi-Quadrat-Verteilung”, “Inverse F-Verteilung”, “Inverse Normalverteilung” und “Inverse T-Verteilung” wurden in LabVIEW 8.0 durch die Instanzen “Chi-Quadrat”, “F”, “Normal” und “Student-t” des VIs “Stetige inverse Verteilung” ersetzt.
- Die VIs “1D-Linearentwicklung” und “2D-Linearentwicklung” sind in LabVIEW 8.0 nicht mehr in der Palette **Funktionen** enthalten. Verwenden Sie stattdessen das VI “Linearentwicklung”.
- Die VIs “1D-Polynomentwicklung” und “2D-Polynomentwicklung” sind in LabVIEW 8.0 nicht mehr in der Palette **Funktionen** enthalten. Verwenden Sie stattdessen das VI “Polynomentwicklung”.
- Die VIs “1D: Kartesisch in polar” und “1D: Polar nach kartesisch” befinden sich in LabVIEW 8.0 nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Re/Im nach Polar” bzw. “Polar nach Re/Im”.
- In LabVIEW 8.0 befindet sich das VI “Frequenzanalyse” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen zur Messung von **Klirrfaktor** und **Oberwellenamplituden** das VI “Klirrfaktoranalyse” bzw. zum Messen der Ausgänge **SINAD** oder **Klirrfaktor plus Rauschen** das VI “SINAD Analyzer”.
- In LabVIEW 8.0 befindet sich das VI “Netzwerkfunktionen (avg)” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die VIs “Übertragungsfunktion (Betrag-Phase)”, “Frequenzantwortfunktion (Real-Im)”, “Kreuzspektrum (Betrag-Phase)” oder “Kreuzspektrum (Real-Im)”.
- In LabVIEW 8.0 befindet sich das VI “Impulsparameter” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen zum Messen von **Anstiegszeit**, **Dauer**, **Überschwingen** oder **Vorschwingen** das VI “Signalvermessung”, zum Messen der Ausgänge **Periode**, **Pulsdauer** oder **Tastverhältnis** das VI “Impulsmessungen” und zum Messen von **Amplitude**, **High-Pegel** oder **Low-Pegel** das VI “Amplitude und Pegel”.
- In LabVIEW 8.0 befindet sich das VI “Übertragungsfunktion” nicht auf der **Funktionen**-Palette. Verwenden Sie stattdessen die VIs “Übertragungsfunktion (Betrag-Phase)” bzw. “Übertragungsfunktion (Real-Im)”.

- In LabVIEW 8.0 befindet sich das Express-VI “NI-DIAdem Report-Assistent” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen das Express-VI “DIAdem-Report”. Weitere Informationen zum Express-VI “DIAdem-Report” finden Sie im Abschnitt *Express-VI “DIAdem-Report”* in diesem Dokument.



Hinweis Für das Express-VI “DIAdem-Report” muss mindestens DIAdem 9.1 Service Pack 2 installiert sein. Zum Herunterladen einer Evaluierungsversion oder zum Kauf von DIAdem besuchen Sie bitte die Website ni.com/info und geben Sie den Infocode `exf5ty` ein.

- In LabVIEW 8.0 befinden sich die Konstanten “VISA-Ressourcenname” und “Logischer IVI-Name” nicht mehr auf der Palette **Funktionen**. Der VISA-Ressourcenname wird nun mit Hilfe des Eingangs **VISA-Ressourcenname** der VISA-VIs und der logische IVI-Name mit dem entsprechenden Eingang des Treiber-VIs festgelegt, mit dem das Gerät initialisiert wird.
- In LabVIEW 8.0 befindet sich die Fehler-Ringkonstante nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen zur Eingabe des gewünschten Fehlercodes eine Konstante für einen vorzeichenlosen 32-Bit-Integer.
- **(Windows)** In LabVIEW 8.0 befinden sich die Audio-VIs, die es in LabVIEW 7.x auf der Palette **Audio** gab, nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Audio-VIs von LabVIEW 8.0. Die mit LabVIEW 7.x gelieferten Beispiele gibt es ebenfalls in LabVIEW 8.0 nicht mehr.
- **(Mac OS und Linux)** Die Audio-VIs, die mit LabVIEW 7.1 ausgeliefert wurden, gibt es auch weiterhin in LabVIEW 8.0. Die mit LabVIEW 7.x gelieferten Beispiele gibt es in LabVIEW 8.0 nicht mehr.

Datei-I/O-VIs und -Funktionen

In LabVIEW 8.0 befindet sich das VI “Zeichen aus Datei lesen” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “Aus Textdatei lesen”.

In LabVIEW 8.0 befindet sich das VI “Öffnen/Erstellen/Ersetzen einer Datei” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “Öffnen/Erstellen/Ersetzen einer Datei”. Die folgenden Funktionen enthalten einen Teil des Funktionsumfangs, den das VI “Öffnen/Erstellen/Ersetzen einer Datei” in LabVIEW 7.x und Vorgängerversionen hatte.

- Die Größe einer Datei lässt sich mit der Funktion “Dateigröße ermitteln” feststellen.

- Der Startpfad, das Dateimuster und der Standardname einer Datei oder eines Verzeichnisses für ein Dateialogfeld werden im Express-VI “Dateialogfeld” festgelegt.
- Eine Refnum lässt sich mit Hilfe der Funktion “Refnum nach Pfad” in einen Pfad umwandeln.
- Mit der Funktion “In Binärdatei schreiben” können Sie plattformunabhängige Textdateien oder andere Arten von Binärdateien erstellen und mit der Funktion “Aus Binärdatei lesen” können Sie die resultierende Binärdatei auslesen.

In LabVIEW 8.0 befinden sich die Funktionen “Datei lesen” und “Datei schreiben” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Aus Binärdatei lesen” und “In Binärdatei schreiben”.

In LabVIEW 8.0 befindet sich das VI “Zeichen in Datei schreiben” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “In Textdatei schreiben”.

In LabVIEW 8.0 befindet sich die Funktion “Zugriffsrechte” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Berechtigungen ermitteln” und “Berechtigungen festlegen”.

In LabVIEW 8.0 befindet sich die Funktion “EOF” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Dateigröße ermitteln” und “Dateigröße festlegen”.

In LabVIEW 8.0 befindet sich die Funktion “Verzeichnis auflisten” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “Ordner anzeigen”.

In LabVIEW 8.0 befindet sich die Funktion “Bereich sperren” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “Zugriff verweigern”.

Wenn Sie ein in LabVIEW 7.x erstelltes VI öffnen und auf dem Blockdiagramm die Funktion “Neues Verzeichnis” enthalten ist, wird diese durch die Funktion “Ordner erstellen” ersetzt. Wenn der von Ihnen im Eingang **Pfad** festgelegte Ordner nicht existiert, erstellt die Funktion “Ordner erstellen” das Verzeichnis. Die Funktion gibt keine Fehlermeldung aus, wie es bei der Funktion “Neues Verzeichnis” der Fall war.

In LabVIEW 8.0 befindet sich die Funktion “Suchen” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Dateiposition ermitteln” und “Dateiposition festlegen”.

In LabVIEW 8.0 befindet sich die Funktion “Datei- und Programmtyp” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen “Datei- und Programmtypen ermitteln” bzw. “Datei- und Programmtypen festlegen”.

In LabVIEW 8.0 befindet sich die Funktion “Datenträger-Info” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion “Datenträger-Info lesen”.

In LabVIEW 8.0 befinden sich die Funktionen “Datei öffnen” und “Neue Datei” nicht mehr auf der Palette **Funktionen**. In LabVIEW 8.0 befindet sich das VI “Zeichen aus Datei lesen” ebenfalls nicht mehr auf der Palette **Funktionen**, wird jedoch aus Gründen der Kompatibilität mitgeliefert.

In LabVIEW 8.0 befinden sich die VIs “Aus I16-Datei lesen”, “Aus Sgl-Datei lesen”, “In I16-Datei schreiben” und “In Sgl-Datei schreiben” nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die VIs “Aus Binärdatei lesen” und “In Binärdatei schreiben”.

Suchen veralteter VIs

Führen Sie folgende Schritte aus, um in den mit früheren LabVIEW-Versionen erstellten VIs nach veralteten VIs zu suchen:

1. Laden Sie die zu durchsuchenden VIs in den Speicher.
2. Klicken Sie zur Anzeige des Dialogfelds **Suchen** auf **Bearbeiten» Suchen und Ersetzen**.
3. Klicken Sie auf die Schaltfläche **Text**.
4. Wählen Sie im Pulldown-Menü **Applikationsinstanz** die Applikation mit den zu durchsuchenden VIs aus. Alles Weitere zu Applikationsinstanzen ist im Abschnitt *Arbeiten mit mehreren Kopien einer Anwendung* beschrieben.
5. Wählen Sie aus dem Pulldown-Menü **Suchbereich** die Option **Alle VIs in Applikationsinstanz**.
6. Aktivieren Sie die Option **Hierarchie-Fenster**.
7. Klicken Sie auf die Schaltfläche **Suchen**.

Änderungen der Funktionsweise von Eigenschaften, Methoden und Ereignissen

Die Funktionsweise folgender Eigenschaften, Methoden und Ereignisse hat sich in LabVIEW 8.0 geändert.

Eigenschaften und Methoden von Applikationen

In LabVIEW 8.0 hängt das Verhalten einiger Applikationseigenschaften und -methoden von der Kopie der Anwendung (Applikationsinstanz) ab, zu der sie gehören. So richtet sich zum Beispiel die Funktionsweise der Eigenschaft “Applikation:Alle VIs im Speicher” von der Applikationsinstanz ab, in der sie sich befindet. Diese Eigenschaft gibt eine Liste mit allen VIs aus, die sich im Speicher derselben Applikationsinstanz befinden wie die Eigenschaft. Dagegen ist die Funktionsweise der Eigenschaft “Applikation:Verzeichnispfad” von der Applikationsinstanz unabhängig. Diese Eigenschaft gibt einen absoluten Pfad zu dem Verzeichnis aus, in dem sich die Applikation befindet. Diese Angabe ist für alle Applikationsinstanzen gleich.

Alles Weitere zu Applikationsinstanzen ist im Abschnitt *Arbeiten mit mehreren Kopien einer Anwendung* beschrieben.

Methode “Frontpanel:Öffnen”

In LabVIEW 7.1 wurde die Methode “FP öffnen” aus LabVIEW 7.0 “FP öffnen - Alt” genannt. LabVIEW 7.1 enthält eine andere Methode “FP öffnen”, die keine Fehlermeldung mehr ausgibt, wenn das Frontpanel bereits geöffnet ist. Die Methode “FP öffnen” aus LabVIEW 7.1 wurde in LabVIEW 8.0 in “Frontpanel:Öffnen” umbenannt. Ersetzen Sie bei VIs, in denen die Methode “FP öffnen - Alt” enthalten ist, diese Methode durch “Frontpanel:Öffnen”.

Ereignisse “Taste gedrückt” und “Tastenwiederholung”

Das Datenfeld **VTaste** der Ereignisse “Taste gedrückt”, “Taste gedrückt?”, “Tastenwiederholung” und “Tastenwiederholung?” für VIs und Bedienelemente gibt nun zwei verschiedene Werte aus, je nachdem, ob die Taste <Return> im Buchstabenblock und die Taste <Enter> im Zahlenblock der Tastatur gedrückt wurde. Wenn in LabVIEW 7.x oder Vorgängerversionen durch Betätigen der <Enter>- oder <Return>-Taste eines dieser Ereignisse ausgelöst wird, gibt LabVIEW im Datenfeld **VTaste** immer <Enter> aus. Wenn in LabVIEW 8.0 durch Betätigen der <Enter>- oder <Return>-Taste eines dieser Ereignisse ausgelöst wird, gibt LabVIEW unter **VTaste** entweder <Enter> oder <Return> aus.

(Mac OS) In LabVIEW 8.0 können Kontextmenüs nur noch durch einen Klick bei gedrückter <Control>-Taste und nicht mehr bei gedrückter <Command>-Taste geöffnet werden. Wenn Sie dieses Verhalten mit einer Ereignisstruktur nachahmen, passen Sie Ihre VIs der neuen Funktionsweise entsprechend an.

Eigenschaften für Listenfeld

Wenn Sie in LabVIEW 7.x oder Vorgängerversionen die Eigenschaft “Erste Zeile” in einem Listenfeld auf eine Zeile einstellen, die sich unter dem untersten Element im Listenfeld befindet, so fixiert LabVIEW die Reihe am letzten sichtbaren Element. In LabVIEW 8.0 hat die Anzahl sichtbarer Einträge keinen Einfluss auf die Anzahl der Zeilen, die mit dieser Eigenschaft verbunden werden können.

LabVIEW 8.0 unterstützt die Eigenschaft “Doppelklick” für einspaltige Listenfelder nicht. Verwenden Sie stattdessen die Methode “Doppelt angeklickte Zeile ermitteln”.

Eigenschaft “Übergeordnetes VI”

In LabVIEW 7.x und Vorgängerversionen gibt die Eigenschaft “Übergeordnetes VI” eine Refnum für das VI aus, zu dem das Objekt gehört. Diese Refnum behält das VI im Speicher. In LabVIEW 8.0 behält die Refnum, die von der Eigenschaft “Übergeordnetes VI” ausgegeben wird, das VI nicht im Speicher. Wenn das übergeordnete VI nicht mehr im Arbeitsspeicher ist, wird die Refnum daher ungültig. Eine Refnum für ein VI, mit der das VI so lange im Speicher verbleibt, bis die Refnum per Hand geschlossen wird, kann mit der Funktion “VI-Referenz öffnen” erzeugt werden.

Texteigenschaft

In LabVIEW 7.x und Vorgängerversionen gibt die Eigenschaft “Text” einen String in normaler Darstellung aus. In LabVIEW 8.0 dagegen gibt die Eigenschaft einen String in derselben Darstellungsart aus wie das Frontpanel-Objekt. Wenn Sie das Frontpanel-Objekt beispielsweise auf Passwortanzeige gestellt haben, gibt die Eigenschaft den Text auch in der Passwortanzeige aus.

Eigenschaften für Baumstruktur-Element

In LabVIEW 7.x und Vorgängerversionen geben “Eigenschaften aktiver Zellen:Zellengröße:Höhe” und “Eigenschaften aktiver Zellen:Zellengröße:Breite” 17 Pixel für jede Zeile im Baumstruktur-Element aus. In LabVIEW 8.0 geben sie 16 Pixel für jede Zeile im Baumstruktur-Element aus.

VI-String-Methoden

Strings, die Sie aus früheren LabVIEW-Versionen mit Hilfe der Methode “VI-Strings exportieren” exportieren, werden mit der Methode “VI-Strings:Importieren” möglicherweise nicht ordnungsgemäß in LabVIEW 8.0 importiert.

Veraltete Eigenschaften, Methoden und Ereignisse

Die nachfolgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW 8.0 nicht mehr unterstützt.

Eigenschaften für Cursor

Die Eigenschaft “Cursor-Fixiermodus” wird in LabVIEW 8.0 nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Cursor-Modus”.

Eigenschaften für Listenfelder, Tabellen, Digitaltabellen, Baumstrukturen und Ereignisse

In LabVIEW 8.0 wird die Eigenschaft “Zellenvordergrundfarbe” für mehrspaltige Listenfelder nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Aktive Zelle:Schriftart der Zelle:Farbe”.

In LabVIEW 8.0 wird die Eigenschaft “Zellenvordergrundfarbe” für Text- und Binärtabellen nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Aktive Zelle:Schriftart für Zelle:Farbe” für Tabellen und Digitaltabellen.

In LabVIEW 8.0 wird die Eigenschaft “Eigenschaften aktiver Zellen: Vordergrundfarbe” für Baumstrukturelemente nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Aktive Zelle:Schriftart der Zelle:Farbe”.

In LabVIEW 8.0 werden die Ereignisse “Ziehen”, “Ziehen?”, “Ablegen” und “Ablegen?” in der Klasse “Baumstruktur” nicht mehr unterstützt. Verwenden Sie stattdessen die Ereignisse “Ziehen beendet”, “Ziehen - Eintritt”, “Ziehen - Austritt”, “Ziehen - Über”, “Ziehen - Quelle aktualisiert”, “Ziehen begonnen”, “Ziehen begonnen?” und “Ablegen” in der Klasse “Bedienelement”.

Eigenschaften für numerischesElementMitText

In LabVIEW 8.0 werden die Eigenschaften “Farben angegebenes NumElement”, “Farben angegebenes NumElement:Hintergrundfarbe” und “Farbe angegebenes numerisches Element:Textfarbe” für numerische Elemente nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaften “Textfarben”, “Textfarben:Hintergrundfarbe” und “Textfarben:Textfarbe”.

Eigenschaften für Panel

In LabVIEW 8.0 wird die Eigenschaft “Farbe” in der Klasse “Frontpanel” nicht mehr unterstützt. In LabVIEW 8.0 gilt die Eigenschaft nur für den linken oberen Fensterbereich. Verwenden Sie stattdessen die Eigenschaft “Farbe des Fensterbereichs” in der Klasse “Fensterbereich”.

Unterschiedliche Eigenschaften

Verwenden Sie in LabVIEW 8.0 das Feld eines SubVIs in einem Unterpanel, um die Anzeige der Bildlaufleisten für Unterpanel-Elemente zu konfigurieren und das Frontpanel in den Unterpanel-Elementen zu skalieren.

In LabVIEW 8.0 wird die Unterpanel-Eigenschaft “X-Bildlaufleiste sichtbar” nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Sichtbarkeit der horizontalen Bildlaufleiste” für den Fensterbereich.

In LabVIEW 8.0 wird die Unterpanel-Eigenschaft “Y-Bildlaufleiste sichtbar” nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Sichtbarkeit der vertikalen Bildlaufleiste” für den Fensterbereich.

In LabVIEW 8.0 wird die Unterpanel-Eigenschaft “Panel skalieren” nicht mehr unterstützt. Verwenden Sie stattdessen die Methode “Skalierungsmodus einstellen” für den Fensterbereich.

Eigenschaften, Methoden und Ereignisse für VIs

In LabVIEW 8.0 wird die Eigenschaft “Frontpanel-Fenster:Autom. zentrieren” nicht mehr unterstützt. Verwenden Sie stattdessen die Methode “Frontpanel:Zentriert”.

In LabVIEW 8.0 wird die Eigenschaft “Frontpanel-Fenster:Bildschirmfüllend anzeigen” nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft “Frontpanel-Fenster:Status”.

In LabVIEW 8.0 wird die Eigenschaft “Frontpanel-Fenster:Ursprung” in dieser VI-Klasse nicht mehr unterstützt. In LabVIEW 8.0 gilt die Eigenschaft nur für den linken oberen Fensterbereich. Verwenden Sie stattdessen die Eigenschaft “Ursprung” in der Klasse “Fensterbereich”.

In LabVIEW 8.0 wird die Eigenschaft “Frontpanel-Fenster:Bildlaufleisten anzeigen” in dieser VI-Klasse nicht unterstützt. In LabVIEW 8.0 gilt die Eigenschaft nur für den linken oberen Fensterbereich. Verwenden Sie stattdessen die Eigenschaften “Sichtbarkeit der horizontalen Bildlaufleiste” und “Sichtbarkeit der vertikalen Bildlaufleiste” in der Klasse “Fensterbereich”.

In LabVIEW 8.0 werden die Methoden “Frontpanel-Skalierungsmodus ermitteln” und “Frontpanel-Skalierungsmodus einstellen” in der VI-Klasse nicht mehr unterstützt. In LabVIEW 8.0 gelten diese Methoden nur für den linken oberen Fensterbereich. Verwenden Sie stattdessen die Methoden “Skalierungsmodus lesen” und “Skalierungsmodus setzen” in der Klasse “Fensterbereich”.

In LabVIEW 8.0 werden die Ereignisse “Maustaste gedrückt”, “Maustaste gedrückt?”, “Maus tritt in Bereich ein”, “Maus verlässt Bereich”, “Mausbewegung” oder “Maustaste losgelassen” in der VI-Klasse nicht mehr unterstützt. Verwenden Sie stattdessen die Ereignisse “Maustaste gedrückt”, “Maustaste gedrückt?”, “Maus tritt in Bereich ein”, “Maus verlässt Bereich”, “Mausbewegung” oder “Maustaste losgelassen” in der Klasse “Fensterbereich”.

Tags der LabVIEW-Standardmenüpunkte

Folgende Applikationselement-Tags wurden aus LabVIEW entfernt:

- APP_BUILD_STANDALONE_APP
- APP_DN_ASSEMBLY_REFS
- APP_EDIT_VI_LIBRARY
- APP_SAVE_WITH_OPTIONS
- APP_SHOW_CLIPBOARD
- APP_SRC_CODE_CTRL
- APP_SWITCH_EXEC_TARGET
- APP_UPDATE_VXI
- APP_VIEW_PRINTED_MANUALS

Bei Laufzeitmenü-Dateien (*.rtm-Dateien), die in einer Vorgängerversion von LabVIEW gespeichert wurden, wird das Tag automatisch aus der *.rtm-Datei entfernt, wenn die Datei im **Menü-Editor** gespeichert wird. Die gelöschten Applikationselement-Tags sind für LabVIEW reserviert und können nicht als Benutzer-Tags verwendet werden.

Unterstützung von HiQ

HiQ wird in LabVIEW 8.0 nicht mehr unterstützt. Wenn in einer Ihrer Applikationen HiQ-VIs vorkommen, sollten Sie diese eventuell durch Mathematik- und Signalverarbeitungs-VIs ersetzen. Informationen zur Verwendung dieser VIs finden Sie in der *LabVIEW-Hilfe*.

Fenster “Fehlerliste”

In LabVIEW 7.x und Vorgängerversionen werden im Abschnitt **VI-Liste** des Fensters **Fehlerliste** alle Fehler in den VIs im Speicher angezeigt. In LabVIEW 8.0 werden in der Fehlerliste unter **Fehlerhafte Elemente** die Fehler an allen Objekten im Speicher, wie VIs oder Bibliotheken, angezeigt. Wenn mehrere Elemente den gleichen Namen haben, wird hier die Applikationsinstanz für jedes Element angezeigt. Alles Weitere zu Applikationsinstanzen ist im Abschnitt *Arbeiten mit mehreren Kopien einer Anwendung* beschrieben.

Syntax für VI-String-Dateien

Wenn mit der Option **Werkzeuge»Fortgeschritten»Strings importieren** oder der Methode “VI-Strings:Importieren” VI-String-Dateien importiert werden, sucht LabVIEW 8.0 nach den Tags `<GROUPER></GROUPER>`. Diese Tags stehen für Frontpanel-Objekte, die zu einer Gruppe zusammengefasst sind. Daher können in LabVIEW 8.0 keine VI-String-Dateien importiert werden, die in älteren Versionen von LabVIEW erstellt wurden.

In LabVIEW 7.1 und Vorgängerversionen werden Strings in einem Listenfeld im Abschnitt `<ITEMS>` der Privatdaten aufgelistet. In LabVIEW 8.0 werden Strings in einem Listenfeld im Abschnitt `<STRINGS>` der Privatdaten aufgelistet. In LabVIEW 7.1 und Vorgängerversionen kann für ein Listenfeld nur eine Schriftart verwendet werden. Diese wird im Abschnitt `<LBLABEL>` der Privatdaten aufgeführt. In LabVIEW 8.0 dagegen können in einem Listenfeld mehrere Schriftarten vorkommen. Diese werden im Abschnitt `<CELL_FONTS>` der Privatdaten aufgelistet.

In LabVIEW 7.1 und Vorgängerversionen werden Strings in einem mehrspaltigen Listenfeld in den Standarddaten aufgelistet. Der Standardwert für ein mehrspaltiges Listenfeld ist jedoch ein Integer oder ein Array aus Integer. In LabVIEW 8.0 werden Strings in einem mehrspaltigen Listenfeld in den Privatdaten aufgelistet.

In LabVIEW 7.1 und Vorgängerversionen werden weder Text noch Schriftarten aus Baumstrukturen exportiert. In LabVIEW 8.0 können sowohl Baumstruktur-Strings als auch -Schriftarten exportiert werden. Diese werden in das gleiche Format exportiert wie bei einem Listenfeld bzw. mehrspaltigen Listenfeld.

In LabVIEW 8.0 enthält jede Zeile einer Exportdatei nicht mehr als zwei Tags für Privat- oder Standarddaten. In LabVIEW 8.0 werden die Elemente auch auf jeder Unterebene eingerückt.

Führen Sie folgende Schritte aus, um VI-String-Dateien in das Format von LabVIEW 8.0 zu konvertieren:

1. Importieren Sie die VI-String-Datei in die vorherige LabVIEW-Version.
2. Speichern Sie das VI.
3. Laden Sie das VI in LabVIEW 8.0.
4. Wählen Sie **Werkzeuge»Fortgeschritten»Strings exportieren**, um die VI-String-Datei im Format von LabVIEW 8.0 zu speichern.

Konvertieren von Typdeskriptordaten in das bzw. aus dem Format von LabVIEW 7.x

Das Speicherformat von Typdeskriptoren hat sich in LabVIEW 8.0 geändert. In LabVIEW 7.x werden Typdeskriptoren in Form von 16-Bit-Strings gespeichert und in LabVIEW 8.0 in Form von 32-Bit-Strings. Dadurch wird die Beschränkung der Größe auf 64 KB für Typdeskriptoren aufgehoben.

In LabVIEW 8.0 ist es allerdings möglich, die in LabVIEW 7.x gespeicherten Typdeskriptoren zu lesen und so zu speichern, dass sie mit LabVIEW 7.x kompatibel sind. Die Funktion "In String konvertieren" hat dazu nun den Kontextmenüpunkt **7.x-Daten konvertieren**. Wenn Sie die Funktion mit der rechten Maustaste anklicken und diesen Menüpunkt auswählen, wird mit den Eingangsdaten so verfahren, als ob in LabVIEW 7.x gearbeitet würde. Wenn Sie die Option **7.x-Daten konvertieren** aus dem Kontextmenü auswählen und am Ausgang **Daten-String** ein String anliegt, wird die Funktion mit einem roten 7.x gekennzeichnet, um anzuzeigen, dass die Daten nicht im üblichen LabVIEW-8.0-Format vorliegen. Sollen die Daten nicht (mehr) umgewandelt werden, wird die Menüoption **7.x-Daten konvertieren** beim erneuten Anklicken deaktiviert.

Wenn in LabVIEW 8.0 ein VI geladen wird, das zuletzt in LabVIEW 7.x oder einer älteren Version gespeichert wurde, wird bei der Funktion "In String konvertieren" die Einstellung **7.x-Daten konvertieren** automatisch aktiviert. Die Funktion arbeitet weiterhin wie in LabVIEW 7.x und Vorgängerversionen. Wenn für ein VI das Typdeskriptorformat von LabVIEW 8.0 verwendet werden soll, klicken Sie mit der rechten Maustaste auf die Funktion "In String konvertieren" und deaktivieren Sie die Option **7.x-Daten konvertieren**. Wenn VIs weder mit Dateien arbeiten, die Daten im LabVIEW-7.x-Format oder einem älteren Format enthalten, noch Daten mit VIs von LabVIEW 7.x oder einer Vorgängerversion austauschen, arbeiten Sie mit dem LabVIEW-8.0-Typdeskriptorformat. Das alte Format wird unter Umständen in zukünftigen Versionen nicht mehr unterstützt.

Umwandeln von NaN-Strings in Integer (Windows)

Wenn NaN in LabVIEW 7.x in einen Integer umgewandelt wird (entweder programmatisch oder durch den Benutzer), stellt dieser Wert den kleinsten Wert für diesen Integer-Datentyp dar. So ergibt die Umwandlung von NaN in einen 16-Bit-Integer mit Vorzeichen beispielsweise -32,768. Das ist der kleinstmögliche Wert für einen 16-Bit-Integer mit Vorzeichen.

Wenn in LabVIEW 8.0 NaN in einen Integer umgewandelt wird, stellt dieser Wert den größten Wert für diesen Integer-Datentyp dar. So ergibt die Umwandlung von NaN in einen 16-Bit-Integer mit Vorzeichen beispiels-

weise 32,767. Das ist der größtmögliche Wert für einen 16-Bit-Integer mit Vorzeichen.

Mit Case-Strukturen verbundene Konstanten

In LabVIEW 7.x und älteren Versionen können SubVIs im Speicher gehalten werden, indem eine Konstante mit einer Case-Struktur verbunden wird und die SubVIs in einem niemals aufgerufenen Case untergebracht werden. Wenn Sie zum Beispiel eine TRUE-Konstante mit einer Case-Struktur verbinden und ein SubVI im FALSE-Case unterbringen, lädt LabVIEW das SubVI zusammen mit dem aufrufenden VI. In LabVIEW 8.0 werden alle nicht benötigten Blockdiagramm-Abschnitte entfernt. Wenn Sie also ein in einer älteren Version gespeichertes VI mit einer Konstante, die mit einer Case-Struktur verbunden ist, in LabVIEW 8.0 laden, wird die Konstante in ein verborgenes Bedienelement umgewandelt, um die Funktionsweise der älteren Version beizubehalten.

Verzögern von Meldungen des Betriebssystems

In LabVIEW 7.x werden Meldungen des Betriebssystems bearbeitet, während Callback-VIs zum Bearbeiten von .NET- und ActiveX-Ereignissen ausgeführt werden. In LabVIEW 8.0 wird die Bearbeitung von Meldungen des Betriebssystems so lange verzögert, bis die Ausführung des Callback-VIs beendet ist bzw. bis Sie ein modales Dialogfeld geladen haben. Durch diese Verzögerung werden Callback-VIs ohne Unterbrechung ausgeführt. Es wird auch kein Ereignis in einem anderen Ereignis ausgelöst, was zu einem Sperrstatus führen kann.

Sie können von einem Callback-VI keine synchronen Aufrufe zu nicht modalen Dialogfenstern starten. Die Aufrufe eines nicht modalen Dialogfensters von einem Callback-VI müssen asynchron sein. Das wird erreicht, indem Sie die Methode "VI ausführen" im Dialogfenster aufrufen und eine boolesche Konstante mit dem Wert FALSE mit dem Eingang **Auf Ende der Operation warten** verbinden.

In LabVIEW 7.x werden Meldungen des Betriebssystems bearbeitet, während DLL- bzw. Shared-Library-Funktionen ausgeführt werden. In LabVIEW 8.0 wird die Bearbeitung von Meldungen des Betriebssystems verzögert, bis die Aufrufe der DLL-Funktionen beendet sind bzw. bis Sie ein modales Dialogfeld von der DLL aus geladen haben. Durch diese Verzögerung werden DLL-Funktionen ohne Unterbrechung ausgeführt. LabVIEW ruft auch nicht die gleiche DLL auf, während eine DLL-Funktion ausgeführt wird, was zu einem Sperrstatus führen kann.

Ist dieses Standardverhalten eingestellt, können keine synchronen Aufrufe nicht modaler Dialogfelder erfolgen, während eine DLL ausgeführt wird. Die Aufrufe eines nicht modalen Dialogfensters von einer DLL aus müssen asynchron sein. Das wird erreicht, indem Sie die Methode “VI ausführen” im Dialogfenster aufrufen und eine boolesche Konstante mit dem Wert FALSE mit dem Eingang **Auf Ende der Operation warten** verbinden.

In DLLs, die Sie selbst erstellen, können Sie bestimmen, ob Meldungen des Betriebssystems verzögert werden sollen oder nicht. Deaktivieren Sie auf der Seite **Fortgeschritten** des Dialogfelds **Meine DLL-Einstellungen** die Option **Betriebssystemmeldungen in DLL verzögern**. Auf diese Weise können Meldungen des Betriebssystems während der Ausführung von DLL-Funktionen bearbeitet werden.

Ressourcenmanager (Mac OS)

LabVIEW 7.x und frühere Versionen bieten undokumentierte Eigenschaften, mit denen Macintosh-Ressourcendateien gelesen und geschrieben werden können. Diese Methoden gibt es in LabVIEW 8.0 jedoch nicht mehr. Hilfsprogramme, in denen diese undokumentierten Eigenschaften genutzt werden, funktionieren nicht mehr. Daher kann nicht mehr mit Macintosh-Ressourcendateien gearbeitet werden.

Dialogfelder mit einer oder zwei Schaltflächen

In LabVIEW 7.x und Vorgängerversionen können VIs zur Anzeige von Dialogfeldern mit einer oder zwei Schaltflächen nicht programmatisch abgebrochen werden. In LabVIEW 8.0 dagegen können Sie ein solches VI programmatisch abbrechen, indem Sie die Methode “VI abbrechen” aufrufen.

Eigenschafts- und Methodenknoten

Wenn Sie in LabVIEW 7.x einen nicht sichtbar mit einem Objekt verbundenen Eigenschafts- oder Methodenknoten von einer Cursor-Legende aus erstellen, wird der Knoten gelöscht, wenn das VI in LabVIEW 8.0 geöffnet wird.

Update von Shared Libraries oder DLLs

Wenn Sie in LabVIEW 7.x oder einer Vorgängerversionen eine Shared Library oder DLL erstellen, die mit `labview.lib` verknüpft ist, verbinden Sie die DLL in LabVIEW 8.0 stattdessen mit `labviewv.lib`. Die Verwendung von DLLs in LabVIEW 8.0 ist im Abschnitt [Verwendung von DLLs in mehreren LabVIEW-Versionen](#) beschrieben.

Seitenränder beim Drucken

In LabVIEW 7.x und Vorgängerversionen werden die Seitenränder in der Option **Seitenränder** der Seite **Drucken** im Dialogfeld **Optionen** in Zentimetern angegeben. In LabVIEW 8.0 werden die **Seitenränder** in Millimetern angegeben.

Upgrade von LabVIEW 6.x

Bei einem Upgrade von LabVIEW 6.x auf LabVIEW 8.0 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Sonstige Probleme, die beim Upgrade von LabVIEW auftreten können, sind im Abschnitt *Upgrade von LabVIEW 7.x* beschrieben.

Alle Unterschiede, die es zwischen den einzelnen Versionen ab 6.x gibt, sind in den *LabVIEW Upgrade Notes* zu jeder Version beschrieben, die Sie unter ni.com/manuals finden.

Änderungen am Datentyp “Signalverlauf”

Der Datentyp “Signalverlauf” verwendet in LabVIEW 7.0 anstelle einer Fließkommazahl mit doppelter Genauigkeit für die Komponente **t0** den Datentyp “Zeitstempel”. Wenn Sie Daten in diesem Datenformat in LabVIEW 6.x speichern, ohne die Änderung in LabVIEW-Versionen ab 7.x zu berücksichtigen, tritt beim Abrufen der Daten unter Umständen ein Fehler auf.

In LabVIEW 7.x und späteren Versionen wandelt das VI “Signalverlauf aus Datei lesen” das alte Signalverlaufsdattentyp-Format in einer Datei in ein neues um. Ein Dialogfeld öffnet sich, in dem Sie aufgefordert werden, die Umwandlung zu akzeptieren. In der LabVIEW Runtime-Engine kann das VI “Signalverlauf aus Datei lesen” diese Umwandlung jedoch nicht vornehmen. Daher wird eine Fehlermeldung ausgegeben. Wenn Sie auf der Website ni.com/info den Infocode `exd9zq` eingeben, erfahren Sie, was es im Hinblick auf Signalverläufe bei der Umstellung von LabVIEW 6.x auf LabVIEW 7.x und spätere Versionen zu beachten gibt.

VIs für serielle Kompatibilität

In LabVIEW 7.x und späteren Versionen wird das VI “Serielle Kompatibilität” nicht mehr auf der Palette **Funktionen** angezeigt. VIs zur Kommunikation mit VXI-Modulen sollten mit VISA-VIs und -Funktionen erstellt werden.

In LabVIEW-Versionen ab 7.x wird zur Kommunikation mit dem seriellen Treiber des Betriebssystems nicht mehr mit dem `serpdrv`-Treiber gearbeitet. LabVIEW enthält kompatible VIs, die auf VISA basieren. Verwenden Sie in neuen Anwendungen zur Steuerung von Geräten mit serieller

Schnittstelle die VIs und Funktionen der VISA-Palette. Alle VIs, die in Vorgängerversionen von LabVIEW enthalten waren und mit seriellen VIs gearbeitet haben, sind auch in LabVIEW 7.1 und späteren Versionen funktionstüchtig.

Wenn in der ursprünglichen LabVIEW-Anwendung Änderungen an der Zuweisung der Portnummern vorgenommen wurden, ist die Änderung anzugeben. Verwenden Sie dazu das VI "Set Serial Alias Ports" im Verzeichnis `labview\vi.lib\Instr_sersup.lib`. Verbinden Sie ein String-Array mit dem Eingang **VISA Aliases** des VIs und geben Sie die Portbezeichnungen ein, die im Eingangsarray verwendet werden. Jedes Element im Array sollte einen Port darstellen. Wenn Sie beispielsweise Port 0 das VISA-Alias "MySerialPort" zugewiesen haben, geben Sie in das Array **VISA Aliases** als erstes Element `MySerialPort` ein. Vor dem VI "Set Serial Alias Ports" muss immer das VI "VISA: Seriellen Port konfigurieren" aufgerufen werden.

Anwendungsbeispiele für die VISA-VIs und -Funktionen zur Steuerung serieller Geräte finden Sie unter `labview\examples\instr\smp1ser1.lib`.

Standardwerte in Schleifen

In LabVIEW 6.0 und Vorgängerversionen werden bei Nicht-Ausführung von For-Schleifen undefinierte Werte ausgegeben. In LabVIEW-Versionen ab 6.1 geben For-Schleifen voreingestellte Werte aus, wenn für die Anzahl der Durchläufe 0 angegeben wird oder ein Eingang mit aktivierter Auto-Indizierung mit einem leeren Array verbunden wird. In diesem Fall wird die Schleife nicht ausgeführt, und für alle nicht automatisch indizierten Ausgabetunnel werden die Standardwerte für den Tunnel-Datentyp ausgegeben.

Lizenzen für netzwerkgesteuerte Frontpanel

Das LabVIEW Full Development System und der Application Builder enthalten eine Lizenz, mit der es einem Client möglich ist, ein Frontpanel über ein Netzwerk zu bedienen. Das LabVIEW Professional Development System wird mit einer solchen Lizenz für fünf Client-Rechner ausgeliefert.

Diese Lizenzen sind auf mehrere Benutzer erweiterbar.

Vergabe mehrerer Ausführungsthreads

In LabVIEW-Versionen ab 7.1 werden zur Ausführung von VIs mehr Threads vergeben als in den Vorgängerversionen. Daher kann es bei der Verarbeitung mehrerer Threads zu Fehlern kommen, wenn ein Funktionsknoten zum Aufruf externer Bibliotheken fälschlicherweise als ablaufinvariant gekennzeichnet wurde, obwohl die DLL eigentlich nicht ablaufinvariant ist. In der *LabVIEW-Hilfe* finden Sie weitere Informationen zum Knoten für den Aufruf externer Bibliotheken und zur Ablaufinvarianz.

Mit dem VI “Threadconfig” in der Bibliothek `labview\vi.lib\Utility\sysinfo.llb` können Sie die Zuweisung von Threads ändern. Sie können die Ablaufinvarianz von VIs deaktivieren, indem Sie **Datei» VI-Einstellungen** wählen, dann in der Pulldown-Liste **Kategorie** auf **Ausführung** klicken und die Option **ablaufinvariante Ausführung** deaktivieren.

Weitere Informationen zum Zuweisen von Threads finden Sie in der *LabVIEW-Hilfe*.

Gerätetreiber

Bei LabVIEW-Versionen ab LabVIEW 7.x ist die LabVIEW-Instrumententreiber-CD nicht im Lieferumfang enthalten. Im Netzwerk für Gerätetreiber von National Instruments unter ni.com/idnet stehen die Treiber jedoch weiterhin zum Herunterladen zur Verfügung. Auf der Treiber-CD von National Instruments sind unter anderem NI-DAQ und NI-VISA enthalten.

Einheiten und Umrechnungsfaktoren

In LabVIEW-Versionen ab 7.x muss die zusätzliche Einheit nach Verwendung der Funktion “Mehrfacharithmetik” nicht mit der Funktion “Einheit konvertieren” entfernt werden.

Die Umrechnungsfaktoren für Einheiten wurden in LabVIEW 7.1 und späteren Versionen den Richtlinien des vom National Institute for Standards and Technology (NIST) veröffentlichten *Guide for the Use of the International System of Units (SI)* weiter angepasst. Außerdem wurde **Kalorie** in **Kalorie (thermisch)** und **Pferdestärke** in **Pferdestärke (elektrisch)** umbenannt. Die Abkürzungen für diese Einheiten haben

sich nicht geändert. Nachfolgend finden Sie alle Änderungen an den Umrechnungsfaktoren für Einheiten zwischen LabVIEW 6.1 und 8.0:

Tabelle 1. Umrechnungsfaktoren für Einheiten

Einheit	Definition in 6.1	Definition in 8.0
Astronomische Einheit (AU)	149.498.845.000 m	149.597.900.000 m
British Thermal Unit (Mittelwert)	1055,79 J	1055,87 J
Elektronenvolt (eV)	1,602e–19 J	1,60217642e–19 J
Foot-Candle	10,764 lx	10,7639 lx
Pferdestärke verglichen mit Pferdestärke (elektrisch)	745,7 W	746 W Die neue Umrechnung ist exakt.
Gallone (imperial)	4,54596 l	4,54609 l
Lichtjahr	9,4605 Pm	9,46073 Pm
Pound Force	4,448 N	4,448222 N
Rod	502,92 cm	5,029210 m
Slug	14,594 kg	14,59390 kg
Unitäre atomare Masseneinheit (u)	1,66057e–27 kg	1,66053873e–27 kg

Eigenschaft “Panel-Aktualisierungen verzögern”

Bei LabVIEW 6.1 und Vorgängerversionen werden Elemente, die sich inzwischen geändert haben, erst wieder aktualisiert, wenn der Wert FALSE an die Eigenschaft übermittelt wird. Wenn diese Eigenschaft in LabVIEW 8.0 den Wert TRUE hat, werden alle Frontpanel-Objekte mit noch ausstehenden Änderungen neu angezeigt und die nächste Aktualisierung der Objekte wird um eine bestimmte Zeit verschoben. In einigen Fällen nimmt die Neudarstellung der Elemente zusätzliche Zeit in Anspruch.

Wertebereiche numerischer Bedienelemente

In LabVIEW 6.1 und Vorgängerversionen haben einige numerische Elemente per Voreinstellung einen Mindestwert von 0,00, einen Höchstwert von 0,00, einen Inkrement-Wert von 0,00 und die Einstellung **Ignorieren** bei einer Bereichsüberschreitung. In LabVIEW 8.0 werden für diese Elemente die Werte des Standardwertebereichs für den jeweiligen Datentyp verwendet.

Typumwandlungspunkte und -definitionen

In den LabVIEW-Versionen ab 6.1 sind in den Verbindungen Angaben über Typdefinitionen enthalten, so dass im Blockdiagramm mehr Typumwandlungspunkte vorliegen können. Wenn Sie ein Typ-Definitionselement mit einem VI oder einem Knoten verbinden, der kein Typ-Definitionsknoten ist, zeigt LabVIEW einen Typumwandlungspunkt an. Typumwandlungspunkte werden auch angezeigt, wenn ein Ausgangsanschluss, der eine Typdefinition hat, mit einer Anzeige verbunden wird, die keine Typdefinition hat. Diese Typumwandlungspunkte zeigen, an welchen Stellen im VI Typdefinitionen nicht einheitlich verwendet werden. Die Ausführungsgeschwindigkeit des VIs wird davon nicht beeinträchtigt.

Um Typdefinitionen in Strings zu konvertieren, verwenden Sie das VI “Daten in String konvertieren”, zu dem Sie in der *LabVIEW-Hilfe* nähere Einzelheiten finden.

Schaltflächenbeschriftungen im Dialogfeld “Datei”

In LabVIEW 6.1 und Vorgängerversionen wird im Fenster der Dateidialog-Funktion die Schaltfläche **Speichern** angezeigt, wenn der Benutzer einen neuen Dateinamen eingeben kann. Andernfalls lautet die Beschriftung der Schaltfläche **Öffnen**. In LabVIEW 8.0 lautet die Schaltflächenbeschriftung im Dateidialogfeld, das vom Express-VI “Dateidialog” angezeigt wird in allen Fällen **OK**. Die Beschriftung lässt sich nachträglich mit Hilfe des Eingangs **Schaltflächenbeschriftung** des Express-VIs “Dateidialog” ändern. Wenn dieses Express-VI in einem anderen VI verwendet wird, überprüfen Sie das VI, um sicherzustellen, dass die neue Standardbeschriftung **OK** auch dem Zweck Ihres VIs gerecht wird.

Die Funktion “Online-Hilfe steuern”

Der Eingang **Pfad zur Hilfedatei** der Funktion “Online-Hilfe steuern” kann jetzt nicht mehr offen gelassen werden. An den Anschluss kann entweder ein Dateiname mit der Erweiterung `.chm` oder `.hlp` übergeben werden oder der vollständige Pfad zur Datei. Wenn Sie nur den Namen einer kompilierten Hilfedatei angegeben haben, sucht LabVIEW im Verzeichnis `labview\help` nach der Datei.

Methode “VI ausführen”

Wenn Sie in LabVIEW 7.1 den Eingang **Ref automatisch schließen** der Methode “VI ausführen” auf TRUE setzen, wird die Refnum nach Beendigung der VI-Ausführung automatisch geschlossen. In LabVIEW 8.0 wird die Refnum auch dann automatisch geschlossen, wenn die Methode einen Fehler ausgibt. Dadurch kann in einem VI, das auf die Referenz angewiesen ist, während der Ausführung ein Fehler auftreten.

Anzeigen des Frontpanels nach dem Laden

Wenn Sie in LabVIEW 8.0 ein VI so einstellen, dass das Frontpanel beim Laden des VIs sofort sichtbar ist, wird das Frontpanel beim Laden durch den VI-Server nicht angezeigt. In diesem Fall muss das Frontpanel mit Hilfe der Methode "Frontpanel:Öffnen" programmatisch geöffnet werden.

Funktion "VI-Referenz öffnen"

Wird in LabVIEW 6.1 und Vorgängerversionen der Eingang **Optionen** der Funktion "VI-Referenz öffnen" offen gelassen, so erstellt LabVIEW ein VI anhand einer Vorlage, sofern sich die Vorlage nicht bereits im Speicher befindet. Befindet sich die Vorlage im Speicher, erstellt LabVIEW für die Vorlage eine Refnum. Wenn Sie in LabVIEW 7.0 oder einer späteren Version mit Hilfe der Funktion "VI-Referenz öffnen" eine Refnum für eine Vorlage erstellen, die sich bereits im Speicher befindet, gibt die Funktion eine Fehlermeldung aus, es sei denn, Sie legen `0x02` im Parameter **Optionen** fest.

Exponentialschreibweise

In LabVIEW-Versionen vor 6.1 wird zur Exponentialschreibweise in einem Formelknoten das Symbol `^` verwendet. In den LabVIEW-Versionen ab 6.1 ist eine Potenzierung mit dem Symbol `**` gekennzeichnet (zum Beispiel `x**y`). Das Symbol `^` steht nun für das bitweise Exklusiv-ODER (XOR).

IVI-Konfigurationsdateiformat

Bei Dateien zum Speichern von IVI-Konfigurationen wird nun zwischen Groß- und Kleinschreibung unterschieden. Wenn in Ihrer Applikation mit logischen, virtuellen oder Treiber-Session-Namen gearbeitet wird, sollten Sie sich vergewissern, dass der in der IVI-Konfigurationsdatei verwendete Name mit dem angegebenen Namen übereinstimmt.

Formular für technische Unterstützung

In LabVIEW 7.x und späteren Versionen wird `techsup.11b` nicht mehr installiert. Bei Problemen oder Fragen bezüglich der Installation, Konfiguration oder Ausführung von Anwendungen besuchen Sie bitte die Website ni.com/support.

Upgrade von LabVIEW 5.x

Bei einem Upgrade von LabVIEW 5.x auf LabVIEW 8.0 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Alle anderen Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten *Upgrade von LabVIEW 6.x* und *Upgrade von LabVIEW 7.x* beschrieben.

In den *Hinweisen zum Upgrade von LabVIEW* zu jeder Version zwischen LabVIEW 5.x und 8.0 und der *Beilage zum Handbuch für LabVIEW 5.1* unter ni.com/manuals werden alle Unterschiede zwischen den einzelnen Versionen aufgeführt.

Konvertieren von Datenprotokolldateien

In LabVIEW 8.0 werden die Typdefinitionen von Datenprotokolldateien überprüft, um festzustellen, ob eine Konvertierung der jeweiligen Datei erforderlich ist. Wenn die Datei mit einer LabVIEW-Version vor 6.0 erstellt wurde oder der Datentyp "Signalverlauf" darin enthalten ist und aus einer Version vor LabVIEW 7.1 stammt, wird die Datei in LabVIEW 8.0 vor dem Auslesen oder Anhängen von Daten umgewandelt. In allen anderen Fällen werden die Dateien beim Auslesen oder Anfügen von Daten nicht konvertiert.

Beim Öffnen von Datenprotokolldateien in LabVIEW 8.0, die in älteren Versionen des Programms erstellt wurden, werden Sie gefragt, ob die jeweilige Datei in das Format von LabVIEW 8.0 umgewandelt werden soll. Wenn Sie mit der Umwandlung einverstanden sind, wird die vorhandene Datei durch eine neue Datei im richtigen Format ersetzt. Wenn die Datei nicht umformatiert werden soll, gibt LabVIEW einen Fehler aus und die Datei wird nicht geöffnet.



Hinweis Wenn Sie alte Daten von LabVIEW 6.1 und Vorgängerversionen weiterhin nutzen möchten, sollten Sie eine Sicherheitskopie der Datenprotokolldateien anlegen. Eine Umformatierung kann nicht rückgängig gemacht werden, so dass die betreffende Datei für LabVIEW 6.1 oder eine Vorgängerversion verloren ist.

Sollen Datenprotokolldateien beim Öffnen automatisch neu formatiert werden, fügen Sie folgende Zeile in die Datei mit den LabVIEW-Einstellungen ein:

```
silentDataLogConvert=True
```

(Mac OS) Fügen Sie folgende Zeile ein:

```
silentDataLogConvert:True
```

(Linux) Fügen Sie folgende Zeile ein:

```
labview.silentDatalogConvert:True
```

Wenn die Dateien nicht automatisch konvertiert werden sollen, wählen Sie `False` als Einstellung.

Verbinden mit dem VI-Server

Von einem LabVIEW-8.0-Client lässt sich keine Verbindung zum VI-Server von LabVIEW 5.x herstellen, da die in LabVIEW 5.x erstellte Applikation bestimmte Aspekte des LabVIEW-8.0-Serverprotokolls nicht erkennt. Es ist jedoch problemlos möglich, eine Verbindung zwischen einem LabVIEW-5.x-Client und dem VI-Server einer LabVIEW 8.0 Anwendung herzustellen.

UDP-Funktionen

In LabVIEW 6.x und späteren Versionen befinden sich die UDP-VIs nicht auf der Palette **Funktionen**, sondern sind als Kompatibilitäts-VIs in der Bibliothek `labview\vi.lib\oldvers\oldvers.lib` zu finden. Verwenden Sie stattdessen für die Kommunikation über Netzwerke die integrierten UDP-Funktionen.

Upgrade von LabVIEW 4.x

Bei einem Upgrade von LabVIEW 4.x auf LabVIEW 8.0 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 5.x](#), [Upgrade von LabVIEW 6.x](#) und [Upgrade von LabVIEW 7.x](#) beschrieben.

In den *Hinweisen zum Upgrade von LabVIEW* zu jeder Version zwischen LabVIEW 4.x und 8.0 und der *Beilage zum Handbuch für LabVIEW 5.1* unter ni.com/manuals werden alle Unterschiede zwischen den einzelnen Versionen aufgeführt.

Umwandlung boolescher Werte für LabVIEW 4.x

Das Format zur Speicherung boolescher Werte hat sich von LabVIEW 4.x auf LabVIEW 5.x geändert. In LabVIEW 4.x werden boolesche Werte in zwei Bytes gespeichert, sofern sie nicht zu einem Array gehören. In diesem Fall speichert LabVIEW 4.x jeden booleschen Wert in einem einzelnen Bit. In LabVIEW 8.0 wird ein boolescher Wert in einem einzigen Byte gespeichert, und zwar unabhängig davon, ob dieser Wert Teil eines Arrays ist oder nicht. Durch diese Änderung können mehr Funktionen mit booleschen Werten arbeiten und Arrays boolescher Werte sind in ihrem Verhalten

Arrays numerischer Werte ähnlicher. Die Änderung im Speicherformat boolescher Werte wirkt sich in LabVIEW 8.0 auf die Datenverarbeitung in Code-Interface-Knoten (CINs) aus, verhält sich allerdings in LabVIEW 8.0 bei existierenden CINs kompatibel.

Wenn in LabVIEW 4.x Binärdaten in eine Datei gespeichert werden, unter denen sich auch boolesche Werte befinden, ist das Format nicht das gleiche wie beim Speichern der Datei in LabVIEW 8.0. In LabVIEW 8.0 gibt es allerdings eine Möglichkeit, in LabVIEW 4.x gespeicherte Binärdaten auszulesen bzw. so zu speichern, dass sie mit LabVIEW 4.x kompatibel sind. Bei den Funktionen "Typenumformung", "In String konvertieren" und "String in Daten konvertieren" gibt es dazu im Kontextmenü einen Menüpunkt mit der Bezeichnung **4.x-Daten konvertieren**. Wenn Sie die Funktion also mit der rechten Maustaste anklicken und diesen Menüpunkt auswählen, wird mit den Daten so verfahren, als ob in LabVIEW 4.x gearbeitet würde. Mit den Funktionen "In Binärdatei schreiben", "Daten in String konvertieren" oder "Typenumformung" lassen sich Daten erzeugen, die für LabVIEW 4.x formatiert sind. Mit den Funktionen "Aus Binärdatei lesen", "String in Daten konvertieren" oder "Typenumformung" können Daten gelesen werden, die für LabVIEW 4.x formatiert sind. Wenn Sie die Menüoption **4.x-Daten konvertieren** auswählen, wird die Funktion mit einem roten 4.x gekennzeichnet, um anzuzeigen, dass die Daten nicht im üblichen LabVIEW-8.0-Format vorliegen. Sollen die Daten nicht (mehr) umgewandelt werden, wird die Menüoption **4.x-Daten konvertieren** beim erneuten Anklicken deaktiviert.

Bei mehreren Dateien mit booleschen Werten kann auch ein VI erstellt werden, mit dem die Dateien in einem Format geöffnet und gespeichert werden, das LabVIEW 8.0 erkennt.

Wenn in LabVIEW 8.0 ein VI geladen wird, das zuletzt in LabVIEW 4.x oder einer älteren Version gespeichert wurde, wird bei den Funktionen "Datei schreiben", "Datei lesen", "Typenumformung", "In String konvertieren" und "String in Daten konvertieren" die Einstellung **4.x-Daten konvertieren** automatisch aktiviert. Die Funktionen arbeiten genau wie zuvor. Wenn das LabVIEW-8.0-Format für boolesche Werte in dem VI verwendet werden soll, klicken Sie mit der rechten Maustaste auf die Funktion und deaktivieren Sie die Option **4.x-Daten konvertieren**. Wenn VIs weder mit Dateien arbeiten, die boolesche Werte im LabVIEW-4.x- oder einem älteren Format enthalten, noch boolesche Werte mit VIs austauschen, die in LabVIEW 4.x oder einer Vorgängerversion ausgeführt werden, sollten Sie das LabVIEW-8.0-Format verwenden. Das alte Format wird unter Umständen in einer der nächsten Versionen abgeschafft.

VIs zur VI-Steuerung

In LabVIEW 5.x und späteren Versionen befinden sich die VIs zur VI-Steuerung nicht auf der Palette **Funktionen**, sondern sind als Kompatibilitäts-VIs in der Bibliothek `labview\vi.lib\utility\victl.lib` enthalten. Verwenden Sie anstelle der VIs zur VI-Steuerung jetzt die VI-Server-Funktion “VI-Referenz öffnen”, den Knoten “Aufruf über Referenz” sowie den Eigenschafts- und Methodenknoten.

Auch einige Fehlercodes, die die VIs zur VI-Steuerung in LabVIEW 8.0 ausgeben, haben sich geändert. Während die VIs bislang einen Fehlercode zwischen 7 und 1000 ausgaben, arbeiten sie nun mit den Codes 1004 und 1003. Wenn ein mit LabVIEW 4.x erstelltes VI auf einen Fehlercode zwischen 7 und 1000 wartet, muss das VI für LabVIEW 8.0 umgeschrieben werden.

DDE-VIs (Windows)

In LabVIEW 5.x und späteren Versionen befinden sich die DDE-VIs nicht auf der **Funktionen**-Palette, sondern sind als Kompatibilitäts-VIs in der Bibliothek `labview\vi.lib\platform\dde.lib` enthalten.

Upgrade von LabVIEW 3.x oder älteren Versionen

Auf der Website ni.com finden Sie Informationen zum Upgrade von LabVIEW 3.x oder einer älteren Version auf die neueste Version von LabVIEW mit Hilfe des *VI Conversion Kit*. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 4.x](#), [Upgrade von LabVIEW 5.x](#), [Upgrade von LabVIEW 6.x](#) und [Upgrade von LabVIEW 7.x](#) beschrieben.

In den *LabVIEW-Aktualisierungshinweisen* zu jeder Version zwischen LabVIEW 3.x und 8.0 und der *Beilage zum Handbuch für LabVIEW 5.1* unter ni.com/manuals werden alle Unterschiede zwischen den einzelnen Versionen aufgeführt.

Änderungen und Neuerungen in LabVIEW 8.0

Die *LabVIEW-Hilfe* gibt einen Einblick in die Grundlagen der Programmierung mit LabVIEW und enthält schrittweise Anleitungen zum Umgang mit dem Programm sowie Hinweise zu den neuen Funktionen von LabVIEW 8.0. Zum Öffnen der *LabVIEW-Hilfe* klicken Sie auf **Hilfe» LabVIEW-Hilfe durchsuchen**.

Weitere Informationen zu den Eigenschaften und Änderungen in LabVIEW 8.0 finden Sie in der Datei `readme.html` im Verzeichnis `labview`.

Aktivieren der LabVIEW-Lizenz

(**Windows**) LabVIEW beruht auf Lizenzaktivierung. Sie müssen eine gültige Lizenz aktivieren, bevor LabVIEW gestartet werden kann. Zur Aktivierung wird die Seriennummer im Lieferpaket benötigt. Eine LabVIEW-Lizenz wird folgendermaßen aktiviert:

- Geben Sie während der Installation die Seriennummer ein und starten Sie am Ende der Installation den Aktivierungsassistenten.
- Starten Sie LabVIEW im Evaluierungsmodus und klicken Sie dann im Dialogfeld **Aktivierungsbeginn** auf **Aktivieren**.
- Wählen Sie während der Ausführung von LabVIEW im Evaluierungsmodus **Hilfe»LabVIEW aktivieren**. Damit die Lizenzaktivierung wirksam wird, muss LabVIEW neu gestartet werden.

Weitere Informationen zu den Lizenzen von LabVIEW finden Sie in den *Versionshinweisen zu LabVIEW 8.0*.

Starten von LabVIEW

Beim Starten von LabVIEW, wird das Fenster **Erste Schritte** angezeigt. Hier können Sie neue VIs und Projekte erstellen, aus den zuletzt verwendeten LabVIEW-Dateien wählen, nach Beispielen suchen und die *LabVIEW-Hilfe* öffnen.

Beim Öffnen oder Erstellen einer Datei wird das Fenster **Erste Schritte** ausgeblendet. Sie können das Fenster jedoch jederzeit über **Anzeigen»Startfenster** wieder öffnen. Das Fenster **Erste Schritte** wird auch angezeigt, wenn Sie alle Frontpanels und Blockdiagramme schließen.

An- und Abmelden in LabVIEW

Sie können sich in der LabVIEW-Umgebung anmelden, wenn im Netzwerk eine Sicherheitsdomain konfiguriert ist. Durch die Anmeldeinformationen wird LabVIEW bei der Kommunikation mit einem Netzwerk-Server identifiziert, der über eine Zugriffskontrollliste verfügt. Sie können sich programmatisch von LabVIEW abmelden bzw. sich in LabVIEW anmelden und die Änderungen an den Benutzerkonten vom Blockdiagramm aus überwachen, indem Sie die Methoden “NI-Sicherheit:Anmelden” und “NI-Sicherheit:Abmelden” sowie das Ereignis “NI-Sicherheit:Benutzeränderung” verwenden.

Domain-Kontomanager

Unter **Werkzeuge»Sicherheit»Domain-Kontomanager** können Sie Domains anlegen und entfernen, die Einstellungen der Domain durchsetzen sowie Benutzer- und Gruppenkonten innerhalb der Domain oder über ein Netzwerk verwalten. Der Domain-Kontomanager hilft besonders beim

Verwalten von Netzwerk-Domains, wenn ein Benutzer nicht vor Ort am Netzwerkrechner anwesend sein kann, auf dem sich die Domain befindet. Das Protokoll zur Identitätsprüfung ist durch integrierte Datenverschlüsselung und Datenintegration gesichert.

Neuerungen am Dialogfeld “Neu”

Klicken Sie auf den Link **Neu** im Fenster **Erste Schritte**, damit das Dialogfeld **Neu** angezeigt wird. Im Dialogfeld **Neu** können Sie beispielsweise neue VIs, Projekte, Projektbibliotheken, benutzerdefinierte Bedienelemente oder Umgebungsvariablen erstellen. Aktivieren Sie die Option **Zu Projekt hinzufügen**, um einem Projekt ein neues Element hinzuzufügen.

In LabVIEW 7.x und Vorgängerversionen können Sie für die Größe des Dialogfelds **Neu** zwischen den Optionen **Kleines Dialogfeld** und **Großes Dialogfeld** wählen. In LabVIEW 8.0 gibt es diese beiden Schaltflächen nicht mehr. Wenn Sie das Dialogfeld allerdings von Hand verkleinern oder vergrößern, wird es beim nächsten Öffnen von LabVIEW in der eingestellten Größe angezeigt.

In LabVIEW 7.x und Vorgängerversionen wird im Dialogfeld **Neu** eine Vorschau des ausgewählten VIs angezeigt. In LabVIEW 8.0 wird im Abschnitt **Beschreibung** des Dialogfelds **Neu** eine Vorschau des Blockdiagramms angezeigt.

Die Liste **Neu erstellen** enthält keine VI-Vorlagen, Vorlagen für globale Variablen oder Bedienelement-Vorlagen aus LabVIEW x und früheren Versionen. Klicken Sie auf dem Frontpanel oder im Blockdiagramm eines VIs oder einer globalen Variable auf **Datei»Speichern unter** und wählen Sie als **Dateityp** die Option **VI-Vorlagen** aus, um ein Element als Vorlage zu speichern. Zum Speichern eines Bedienelements als Vorlage klicken Sie auf dem Frontpanel eines VIs oder eines benutzerdefinierten Bedienelements auf **Datei»Speichern unter** und wählen Sie **Bedienelement-Vorlagen** als **Dateityp** aus.

RT-, FPGA- und PDA-Systeme

Für die Übertragung von VIs auf RT-, FPGA- oder PDA-Systeme muss ein LabVIEW-Projekt erstellt werden. Weitere Informationen zur Arbeit mit Projekten mit dem RT-, FPGA- und PDA-Modul für LabVIEW finden Sie in der Dokumentation zu den einzelnen Modulen.

Wechseln des Ausführungssystems

Wenn Sie in LabVIEW 7.x und Vorgängerversionen ein Zielsystem auswählen möchten, um dafür VIs zu entwickeln, muss dazu im Dialogfeld **LabVIEW** unter **Ausführungsziel** ein System ausgewählt werden. Wenn Sie gerade an einem VI arbeiten und Sie das System bzw. Gerät ändern

möchten, auf dem das VI laufen soll, wählen Sie unter **Ausführen» Ausführungsziel wechseln** das betreffende System aus. In LabVIEW 8.0 können Sie mit Hilfe der Projektumgebung auf verschiedenen Systemen arbeiten. Klicken Sie mit der rechten Maustaste im **Projekt-Explorer** auf die höchste Ebene und wählen Sie **Neu»Ziele und Geräte**, um zum Dialogfeld **Ziele und Geräte hinzufügen** zu gelangen. Wenn ein System im Projekt andere Systeme unterstützt, können Sie es auch mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Neu»Ziele und Geräte** auswählen. Das neue System wird dann unter dem bestehenden aufgeführt. Sie können unter diesem System ein VI hinzufügen und daran arbeiten. Nach Fertigstellung des VIs können Sie das ausführende System ändern, indem Sie zwischen den VIs unter diesem Zielsystem hin- und herwechseln. Weitere Informationen zum Wechseln des Systems für das LabVIEW-RT-Modul, FPGA-Modul und PDA-Modul finden Sie in der Dokumentation zu den einzelnen Modulen.

LabVIEW-Projekte

Mit Hilfe von Projekten können Sie LabVIEW-Dateien und andere Dateien in Gruppen ordnen, Build-Spezifikationen erstellen, Dateien herunterladen und auf andere Zielsysteme übertragen. Ein Zielsystem ist ein Gerät oder ein Computer, auf dem ein VI ausgeführt wird. Beim Speichern eines Projekts wird eine Projektdatei (*.lvproj) mit den Angaben zur Konfiguration, zum Build und zur Übertragung auf andere Systeme sowie mit Verweisen auf Dateien im Projekt erstellt.

Zum Erstellen von ausführbaren Anwendungen und DLLs muss ein Projekt angelegt werden. Zur Arbeit mit einem RT-, FPGA- oder PDA-System ist ebenfalls ein Projekt anzulegen. Weitere Informationen zur Verwendung von Projekten mit dem LabVIEW-RT-Modul, FPGA-Modul und PDA-Modul finden Sie in der Dokumentation zu den einzelnen Modulen.

Für die LabVIEW-Plug-and-Play-Gerätetreiber zur Projektarbeit werden die Projekt- und Projektbibliotheksfunktionen von LabVIEW 8.0 benötigt. Sie können Treiber für Projekte genauso verwenden wie frühere Plug-and-Play-Treiber von LabVIEW.

Projekt-Explorer

Mit Hilfe des Projekt-Explorers lassen sich Projekte erstellen und bearbeiten. Klicken Sie zum Öffnen des Fensters **Projekt-Explorer** auf **Datei» Neues Projekt**. Sie können auch auf **Projekt»Neues Projekt** oder **Datei» Neu** klicken und dann im Dialogfeld **Neu** die Option **Leeres Projekt** auswählen.

Der Projekt-Explorer enthält per Voreinstellung folgende Komponenten:

- **Höchste Projektebene**—Enthält alle anderen Elemente im Fenster **Projekt-Explorer**. Die höchste Ebene des Projekts ist immer mit dem Dateinamen des Projekts beschriftet.
 - **Mein Computer**—Stellt den lokalen Computer als ein Zielsystem im Projekt dar.
 - **Abhängigkeiten**—Enthält Objekte, die VIs in einem Zielsystem benötigen.
 - **Build-Spezifikationen**—Enthält Build-Konfigurationen für Quellcodepakete und andere Arten von Builds, die es in LabVIEW-Toolkits und -Modulen gibt. Wenn Sie das LabVIEW Professional Development System oder den Application Builder installiert haben, können Sie mit Hilfe der **Build-Spezifikationen** EXE-, DLL- und ZIP-Dateien konfigurieren. (**Windows**) Sie können mit den **Build-Spezifikationen** auch Installationsprogramme konfigurieren.

Wenn Sie dem Projekt ein weiteres System hinzufügen, wird im Projekt-Explorer ein zusätzliches Element erstellt, das für das System steht. Zu jedem System gibt es **Abhängigkeiten** und **Build-Spezifikationen**. Sie können jedem System Dateien hinzufügen.

Sie können ein VI aus dem Projekt-Explorer auf das Blockdiagramm eines anderen geöffneten VIs ziehen und dieses VI dann als SubVI verwenden.

Verwenden eines Projekts mit bereits vorhandenen Dateien

Sie können einem Projekt auch mehrere bereits vorhandene Dateien hinzufügen. Wählen Sie dazu eine der folgenden Optionen aus:

- Wenn die Dateien zu einer Anwendung gehören, wählen Sie **Werkzeuge»Build-Skript konvertieren**, um die Anwendung in ein Projekt umzuwandeln.
- Wenn die vorhandenen Dateien nicht Teil einer Anwendung sind, erstellen Sie ein neues Projekt und fügen Sie die Dateien dem Projekt-Explorer hinzu.

Symbolleisten im Projekt-Explorer

Mit Hilfe der Schaltflächen in den Symbolleisten Standard, Projekt, Build und Versionsverwaltung können Sie Operationen in einem Projekt ausführen. Die Symbolleisten befinden sich ganz oben im Projekt-Explorer. Zur Anzeige aller Symbolleisten müssen Sie möglicherweise die Größe des Fensters **Projekt-Explorer** verändern.

Die Symbolleisten lassen sich ein- bzw. ausblenden, indem Sie auf **Anzeigen»Symbolleisten** klicken und die Symbolleiste auswählen, die ein- bzw. ausgeblendet werden soll. Sie können aber auch eine freie Fläche in der Symbolleiste mit der rechten Maustaste anklicken und die Symbolleisten auswählen, die ein- oder ausgeblendet werden sollen.

Hinzufügen von Objekten zu einem Projekt

Mit Hilfe des Projekt-Explorers lassen sich einem System in einem Projekt sowohl LabVIEW-Dateien wie VIs oder LLBs als auch andere Dateien wie Text- oder Tabellenkalkulationsdateien hinzufügen. Sie können eine Ordnungsstruktur für Objekte in einem Projekt erstellen.

Objekte können einem Zielsystem in einem Projekt auf folgende Art hinzugefügt werden:

- Klicken Sie ein System oder einen Ordner unter einem System mit der rechten Maustaste an, wählen Sie **Datei hinzufügen** aus dem Kontextmenü und wählen Sie die Datei, die hinzugefügt werden soll, aus dem Dialogfeld aus. Sie können jedoch auch erst das System auswählen, dann auf **Projekt»Zu Projekt hinzufügen»Datei hinzufügen** klicken und dann die Dateien aus dem Dateialogfeld auswählen, die hinzugefügt werden sollen.
- Zum Hinzufügen eines leeren VIs klicken Sie mit der rechten Maustaste auf ein Zielsystem und wählen Sie im Kontextmenü **Neu»VI**. Stattdessen können Sie auch **Datei»Neues VI** oder **Projekt»Zu Projekt hinzufügen»Neues VI** auswählen.
- Markieren Sie das VI-Symbol in der rechten oberen Ecke des Frontpanels oder Blockdiagramms und ziehen Sie das Symbol unter das System.
- **(Windows und Mac OS)** Wählen Sie eine Datei oder ein Verzeichnis aus dem Dateisystem und verschieben Sie die Datei oder das Verzeichnis unter das System.

Sie können jedoch einem Projekt auch neue LabVIEW-Dateien aus dem Dialogfeld **Neu** hinzufügen. Wählen Sie dazu **Datei»Neu** oder **Projekt»Zu Projekt hinzufügen»Neu**, um zum Dialogfeld **Neu** zu gelangen. Wählen Sie im Dialogfeld **Neu** das Objekt aus, das hinzugefügt werden soll und aktivieren Sie die Option **Zu Projekt hinzufügen**. Wenn mehrere Projekte geöffnet sind, wählen Sie unter **Projekte** das Projekt aus, dem das Objekt hinzugefügt werden soll.

Hinzufügen von Ordnern zu einem Projekt

Mit Hilfe des Projekt-Explorers können Sie ein Verzeichnis auf einem Laufwerk als Ordner zu einem Projekt hinzufügen. Klicken Sie zum Hinzufügen der Dateien von einem Verzeichnis auf einem Laufwerk mit der rechten Maustaste auf ein Zielsystem oder einen Ordner unter dem System und wählen Sie aus dem Kontextmenü **Ordner hinzufügen** aus. Wenn Sie ein Verzeichnis auswählen, spiegelt das Projekt den gesamten Inhalt des Verzeichnisses einschließlich der Unterverzeichnisse wider. LabVIEW erstellt einen neuen virtuellen Ordner im Projekt mit dem Namen des Verzeichnisses auf dem Datenträger.

Sie können aber auch das System auswählen und dann auf **Projekt»Zu Projekt hinzufügen»Ordner hinzufügen** klicken, um die Dateien aus einem Verzeichnis hinzuzufügen.



Hinweis Wenn Sie einem Projekt ein Verzeichnis von einem Datenträger hinzufügen, wird der entsprechende Ordner im Projekt nicht automatisch aktualisiert, wenn im Verzeichnis auf dem Datenträger Änderungen vorgenommen werden. Wenn Sie an der Struktur der Unterverzeichnisse auf dem Laufwerk Änderungen vornehmen, werden die Unterordner im Projekt nicht automatisch angepasst. Wenn Sie möchten, dass die Ordner in einem Projekt den Verzeichnissen auf dem Laufwerk entsprechen, müssen die Ordner manuell angepasst werden.

Sie können aber zum Ordnen von Objekten in einem Projekt auch neue Ordner erstellen. Klicken Sie mit der rechten Maustaste auf ein System und wählen Sie aus dem Kontextmenü **Neu»Ordner**, um einen neuen Ordner unter dem System hinzuzufügen. Sie können auch einen neuen Unterordner erstellen, indem Sie einen bereits vorhandenen Ordner mit der rechten Maustaste anklicken und im Kontextmenü **Neu»Ordner** wählen.

Klicken Sie mit der rechten Maustaste auf die höchste Ebene im Projekt und wählen Sie aus dem Kontextmenü **Neu»Ordner** aus, um zum Anordnen der Systeme in einem Projekt einen neuen Ordner zu erstellen.

Hinzufügen von LLBs zu einem Projekt

Das Hinzufügen einer LLB zu einem Projekt ist ordner- oder dateiweise möglich. Wenn Sie eine LLB als Ordner hinzufügen, wird der Dateiname der LLB als Bezeichnung für den Ordner übernommen und die VIs der LLB sind im Ordner enthalten. Klicken Sie ein System oder einen Ordner unter einem System mit der rechten Maustaste an, wählen Sie aus dem Kontextmenü **Ordner hinzufügen** aus und wählen Sie die gewünschte LLB aus. Das Hinzufügen oder Entfernen von Objekten aus dem Projekt-Explorer hat keinerlei Auswirkungen auf die *.llb-Datei auf dem Datenträger. Zum Bearbeiten einer LLB klicken Sie auf **Werkzeuge»LLB-Manager**.

Sie können aber auch einem Projekt eine LLB als Datei hinzufügen. Wenn Sie eine LLB als Datei hinzufügen, werden die VIs in der LLB nicht im Projekt-Explorer angezeigt. Klicken Sie mit der rechten Maustaste auf das System oder einen Ordner unter dem System, wählen Sie **Datei hinzufügen** und wählen Sie aus dem Dateidialogfeld die gewünschte LLB aus. Klicken Sie auf das geöffnete Ordnersymbol mit dem Namen **VI**, das rechts mit einem Punkt gekennzeichnet ist, und klicken Sie dann auf die Schaltfläche **Auswahl**.

Entfernen von Projektbestandteilen

Mit Hilfe des Projekt-Explorers lassen sich Objekte aus einem Projekt entfernen. Gehen Sie dazu wie folgt vor:

- Klicken Sie mit der rechten Maustaste auf das Objekt, das entfernt werden soll und wählen Sie aus dem Kontextmenü die Option **Entfernen** aus.
- Wählen Sie das zu löschende Objekt und drücken Sie die <Entf>-Taste.
- Wählen Sie das zu löschende Objekt aus und klicken Sie auf die Schaltfläche **Löschen** in der **Standard**-Symbolleiste.



Hinweis Beim Entfernen eines Projektbestandteils wird dieser nicht vom Datenträger gelöscht.

Anzeigen von Abhängigkeiten in einem Projekt

Unter **Abhängigkeiten** werden Objekte angezeigt, die von den VIs unter einem Zielsystem benötigt werden. Es gibt zu jedem System **Abhängigkeiten**.

Objekte können den **Abhängigkeiten** nicht direkt hinzugefügt werden. Abhängigkeiten für VIs unter einem System werden hinzugefügt, wenn Sie **Abhängigkeiten** mit der rechten Maustaste anklicken und aus dem Kontextmenü **Aktualisieren** auswählen. Wenn Sie zum Beispiel einem System ein VI hinzufügen, das ein SubVI enthält, wird das SubVI auch zu **Abhängigkeiten** hinzugefügt, wenn Sie **Aktualisieren** auswählen. Wenn Sie jedoch ein abhängiges Objekt zu einem System hinzufügen, wird es nicht unter **Abhängigkeiten** angezeigt. Wenn Sie zum Beispiel das VI und das SubVI zum System hinzufügen, wird das SubVI nicht unter **Abhängigkeiten** aufgeführt, wenn Sie **Aktualisieren** auswählen.

Zu Abhängigkeiten gehören VIs, DLLs und Projektbibliotheken, die ein VI statisch aufruft.



Hinweis Objekte, die von einem VI dynamisch aufgerufen werden, werden nicht unter dem Knoten **Abhängigkeiten** angezeigt. Diese Objekte müssen einem System erst hinzugefügt werden, um sie in einem Projekt verwalten zu können.

SubVIs werden von LabVIEW rekursiv überprüft. DLLs dagegen werden nicht rekursiv überprüft. Wenn zum Beispiel a.vi die Datei b.dll statisch aufruft und b.dll die Datei c.dll statisch aufruft, wird nur b.dll als ein abhängiges Objekt betrachtet. Damit c.dll im Projekt verwaltet werden kann, muss c.dll unter dem Zielsystem eingefügt werden.

Ist ein abhängiges Objekt Teil einer Projektbibliothek, wird die gesamte Projektbibliothek zu **Abhängigkeiten** hinzugefügt.

Neue Objekte können nicht direkt unter **Abhängigkeiten** erstellt werden. Sie können keine Objekte von anderen Stellen im Projekt-Explorer in die Kategorie **Abhängigkeiten** verschieben.

Zum Entfernen von Objekten aus **Abhängigkeiten** klicken Sie mit der rechten Maustaste auf ein Objekt und wählen Sie **Entfernen**. Wenn Sie ein Objekt mit dieser Option entfernen, wird es wieder angezeigt, wenn Sie **Abhängigkeiten** mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Aktualisieren** wählen.

Beim Speichern von Projekten werden die Abhängigkeiten nicht als Teil des Projekts gespeichert. Beim Öffnen eines Projekts müssen Sie mit der rechten Maustaste auf **Abhängigkeiten** klicken und aus dem Kontextmenü **Aktualisieren** auswählen, damit die Abhängigkeiten angezeigt werden.

Hinzufügen von Zielsystemen zu einem Projekt (Windows)

Mit Hilfe des Dialogfensters **Ziele und Geräte hinzufügen** lassen sich einem Projekt Zielsysteme und Geräte hinzufügen.



Hinweis Zur Anzeige dieses Dialogfelds ist ein Modul oder Treiber erforderlich, das bzw. der auf dem Zielsystem arbeitet.

Klicken Sie mit der rechten Maustaste auf die höchste Projektebene und wählen Sie **Neu>Ziele und Geräte** aus dem Kontextmenü, um zum Dialogfeld **Ziele und Geräte hinzufügen** zu gelangen. Wenn ein System im Projekt andere Systeme unterstützt, können Sie es auch mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Neu>Ziele und Geräte** auswählen. Das neue System wird dann unter dem bestehenden aufgeführt. Wenn sich zum Beispiel in Ihrem Rechner eine PCI-Karte von National Instruments befindet, können Sie die Karte unter **Mein Computer** hinzufügen.

Übertragen von Dateien auf Zielsystem

Zum Übertragen von Dateien auf ein bestimmtes System muss ein VI im Projekt-Explorer geöffnet und unter dem Zielsystem ausgeführt werden. Während das VI ausgeführt wird, überträgt LabVIEW die Quelldateien, Build-Ergebnisse und die Hardwarekonfiguration auf das Zielsystem.

Verwenden von Paletten mit mehreren Zielsystemen

LabVIEW lädt für jedes Zielsystem separate Paletten. Wenn ein VI auf mehreren Systemen in einem Projekt geöffnet ist, sind auch mehrere Paletten im Speicher vorhanden. Wenn Sie zwischen Systemen wechseln, zeigt LabVIEW jede fixierte Palette auf der entsprechenden Palette im aktiven System an. Wenn es für eine fixierte Palette keine entsprechende Palette gibt, zeigt LabVIEW die übergeordnete Palette für das aktive System an.

Arbeiten mit mehreren Kopien einer Anwendung

LabVIEW erstellt für jedes Ziel in einem Projekt eine Kopie der Anwendung, das heißt, eine Kopie von LabVIEW (Applikationsinstanz). Wenn Sie ein VI vom Projekt-Explorer aus öffnen, wird das VI in der Applikationsinstanz für das Zielsystem geöffnet. Darüber hinaus unterhält LabVIEW eine Art Hauptkopie der Anwendung für geöffnete VIs, die nicht Teil eines Projekts sind bzw. VIs, die nicht von einem Projekt aus geöffnet wurden.

Der Name der Applikationsinstanz, die in der linken unteren Ecke des Frontpanels bzw. des Blockdiagramms angezeigt wird, gibt an, zu welcher Kopie das VI gehört. Wenn mehrere Projekte geöffnet sind, enthält der Name der Applikationsinstanz neben dem Projektnamen auch den Namen des Systems, zum Beispiel **Projekt 1.lvproj/Mein Computer**. Ansonsten enthält der Name der Anwendungskopie nur den Namen des Zielsystems.

Wenn Sie ein geöffnetes VI aus dem Projekt-Explorer entfernen, verbleibt das geöffnete VI in derselben Applikationsinstanz. Wenn Sie das VI schließen und wieder öffnen, wird das VI in der aktuellen Applikationsinstanz geöffnet.

Bearbeiten von VIs in mehreren Applikationsinstanzen

Ein VI auf einem Laufwerk kann gleichzeitig in mehreren Applikationsinstanzen geöffnet werden. Dasselbe VI kann zum Beispiel in zwei verschiedenen Projekten oder auf zwei verschiedenen Systemen innerhalb eines Projekts geöffnet werden.

Wenn Sie ein VI bearbeiten, das nur in einer Applikationsinstanz geöffnet wurde, und dasselbe VI in einer anderen Applikationsinstanz öffnen, enthält die Kopie des VIs, die Sie gerade geöffnet haben, bereits die neuesten Änderungen der bearbeiteten VI-Kopie. Wenn Sie jedoch ein VI bearbei-

ten, das in mehreren Applikationsinstanzen geöffnet ist, werden die Änderungen nicht automatisch von den VIs in den anderen Kopien übernommen. Sie können ein VI erst in anderen Applikationsinstanzen bearbeiten, ausführen oder speichern, wenn die VIs in allen Kopien identisch sind.

Auf folgende Art und Weise werden die VIs in allen Applikationsinstanzen vereinheitlicht:

- Klicken Sie auf die Schaltfläche **Mit anderen geöffneten Anwendungen synchronisieren** in der VI-Symbolleiste, um die Änderungen in den VIs in allen Kopien der Anwendung zu übernehmen.
- Speichern Sie das VI in der Anwendung, in der die Änderungen enthalten sind. Beim Speichern des VIs werden die Änderungen automatisch von den VIs in allen anderen Kopien der Anwendung übernommen.
- Machen Sie die Änderungen am VI in der Anwendung rückgängig.



Hinweis Änderungen am VI können nicht mehr rückgängig gemacht werden, nachdem das VI gespeichert oder synchronisiert wurde.

Ein VI kann nicht bearbeitet werden, wenn es ausgeführt wird oder für die Ausführung in einer anderen Kopie der Anwendung reserviert ist. Gehört ein VI zu einer Projektbibliothek, wird auch die Projektbibliothek in allen Kopien der Anwendung zeitweise blockiert, während das VI ausgeführt wird. Die Projektbibliothek kann nicht bearbeitet werden, während das VI ausgeführt wird. Erst nachdem das VI angehalten wurde, wird die Blockierung der Projektbibliothek aufgehoben.

Verwendung des Webservers für VIs in Projekten

Mit Hilfe des Webservers kann ein VI oder ein Frontpanel in einem Webbrowser angezeigt werden, das Teil eines Projekts ist. Wenn Sie **Ausführen»Mit Netzwerkpanel verbinden** auswählen, fügen Sie den Projektnamen, die Projektbibliothek und das Zielsystem (sofern vorhanden) in das Feld **VI-Name** ein.

Wenn also beispielsweise das VI `Brechungswinkel.vi` zum Projekt `Optikprojekt.lvproj` auf dem System `Physik` gehört, geben Sie den **VI-Namen** so an: `Optikprojekt.lvproj/Physik/Brechungswinkel.vi`. Gehört das VI zur Projektbibliothek `Experimente`, ist diese auch im Feld **VI-Name** anzugeben, also `Optikprojekt.lvproj/Physik/Experimente:Brechungswinkel.vi`. Wenn sich das VI weder in einem Projekt befindet, noch zu einer Projektbibliothek gehört, braucht nur der **VI-Name** angegeben zu werden.

LabVIEW-Projektbibliotheken

LabVIEW-Projektbibliotheken sind eine Zusammenstellung von VIs, Typdefinitionen, Umgebungsvariablen, Menüdateien der Palette und anderen Dateien einschließlich anderer Projektbibliotheken. Wenn Sie eine neue Projektbibliothek erstellen und speichern, erzeugt LabVIEW eine Projektbibliotheksdatei (*.lvlib), in der die Eigenschaften der Projektbibliothek und Referenzen auf die Dateien, die zur Projektbibliothek gehören, enthalten sind.

Projektbibliotheken eignen sich für verschiedene Aufgaben, zum Beispiel zum Anordnen von Dateien auf einer Hierarchieebene, Vermeiden doppelter VI-Namen, Beschränken des offiziellen Zugriffs auf bestimmte Dateien, Beschränken der Berechtigung zum Bearbeiten verschiedener Dateien und Festlegen eines Standardpalettenmenüs für eine Gruppe von VIs.

Die Struktur einer Projektbibliothek kann im Projekt-Explorer oder im Fenster einer eigenständigen Projektbibliothek angezeigt werden. Ist der Projekt-Explorer nicht geöffnet, öffnen Sie diesen über eine Projektbibliotheksdatei im Projektbibliotheksfenster.

Sie können eine Projektbibliothek in einem LabVIEW-Projekt erstellen. Klicken Sie mit der rechten Maustaste auf **Mein Computer** und wählen Sie aus dem Kontextmenü die Option **Neu»Bibliothek** aus. LabVIEW erstellt eine Projektbibliotheksdatei, die unter **Mein Computer** angezeigt wird.

Sie können auch Projektbibliotheken von Ordnern in einem Projekt erstellen.

Anhand von Projektbibliotheken lässt sich eine virtuelle logische Hierarchie von Elementen erstellen. Eine Projektbibliotheksdatei enthält im Vergleich zu einer LLB (eine physikalische Datei, die VIs enthält) nicht die tatsächlichen Dateien. Dateien einer Projektbibliothek werden nach wie vor einzeln auf dem Datenträger in den entsprechenden Verzeichnissen angezeigt. Eine Projektbibliothek kann anders strukturiert sein als die darin enthaltenen Dateien auf dem Datenträger.

Prüfen Sie mit Hilfe von Projektbibliotheken die Namen von VIs und anderen LabVIEW-Dateien. LabVIEW erkennt VIs nach Dateinamen. Wenn ein VI in den Speicher geladen wird, das genauso heißt wie ein anderes VI, arbeitet LabVIEW mit dem VI aus dem Speicher. Dieses Problem wird als Querverlinkung (*Cross-Linking*) bezeichnet. Wenn ein VI Teil einer Projektbibliothek ist, stimmt LabVIEW den VI-Namen mit dem Namen der Projektbibliothek ab, um eine Querverlinkung zu vermeiden. Ein "qualifizierter" Dateiname umfasst den Namen der Datei und den Namen der entsprechenden Projektbibliothek.

Wenn Sie beispielsweise ein VI mit dem Namen `Aufrufer.vi` erstellen, das ein SubVI namens `init.vi` enthält, welches wiederum zu `Bibliothek1.lvlib` gehört, können Sie ein weiteres SubVI namens `init.vi` hinzufügen, wenn dieses zu `Bibliothek2.lvlib` gehört. In diesem Fall tritt keine Querverlinkung auf. Wenn Sie `Aufrufer.vi` speichern, legt LabVIEW `Bibliothek1.lvlib:init.vi` und `Bibliothek2.lvlib:init.vi` als qualifizierte Dateinamen fest.



Vorsicht! Zum Umbenennen einer Projektbibliothek klicken Sie sie mit der rechten Maustaste an und wählen Sie **Umbenennen**. Das Umbenennen einer Projektbibliothek außerhalb von LabVIEW kann zu Fehlern führen.

Klicken Sie mit der rechten Maustaste auf eine Projektbibliothek und wählen Sie **Eigenschaften** aus dem Kontextmenü, um den öffentlichen Zugriff auf bestimmte Arten von Dateien bzw. die Bearbeitungserlaubnis für Projektbibliotheken zu beschränken und um die Palettenmenüdatei (`*.mnu`), die zur Projektbibliothek gehört, als Voreinstellung für alle zur Projektbibliothek gehörenden VIs festzulegen.

Nach Auswahl eines Standardpalettenmenüs können Sie mit der rechten Maustaste auf ein SubVI klicken, ein beliebiges VI der Projektbibliothek aufrufen und die Standardpalette für diese Projektbibliothek aus dem Kontextmenü anzeigen.

Übertragung aktueller Frontpanel-Werte mit Umgebungsvariablen

In LabVIEW 7.x und Vorgängerversionen werden anhand von lokalen und globalen Variablen Daten zwischen Anwendungen ausgetauscht, bei denen keine direkte Verbindung im Blockdiagramm möglich ist. Der Variablentyp kann in Abhängigkeit des Bereichs der Anwendung ausgewählt werden. In LabVIEW 8.0 wird eine Umgebungsvariable eingeführt, mit deren Hilfe Daten zwischen VIs in einem Projekt oder in einem Netzwerk ausgetauscht werden können.

In Umgebungsvariablen ist der gesamte Funktionsumfang der vorhandenen LabVIEW-Datentransfertechnologie (zum Beispiel DataSocket) vereint und Umgebungsvariablen können auch im Projekt-Explorer verwendet werden. Daten können sowohl vom Frontpanel als auch vom Blockdiagramm aus mit Umgebungsvariablen ausgetauscht werden.

Umgebungsvariablen sind konfigurierte Software-Objekte zur Übertragung von Daten zwischen VIs. Mit Hilfe von Umgebungsvariablen können Sie Daten zwischen VIs und zwischen Speicherorten in einer Applikation austauschen, bei denen keine direkte Blockdiagramm-Verbindung möglich ist. Eine Umgebungsvariable kann sowohl eine Einstellung als auch einen Wert darstellen. Sie können den Bereich, den **Variablentyp** oder jede beliebige andere Eigenschaft der Variable ändern, ohne das Blockdia-

gramm des VIs, in dem die Umgebungsvariable verwendet wird, bearbeiten zu müssen.



Hinweis Sie müssen die Firewalls und die NAT-Router (*Network Address Translating*) konfigurieren, wenn Umgebungsvariablen durch Firewalls oder Router übertragen werden sollen.

Erstellen von Umgebungsvariablen

Zum Erstellen einer Umgebungsvariable muss ein Projekt geöffnet sein. Zum Hinzufügen einer Umgebungsvariable zu einem Projekt klicken Sie im Projekt-Explorer mit der rechten Maustaste auf ein System, eine Projektbibliothek oder einen Ordner innerhalb der Projektbibliothek und wählen Sie aus dem Kontextmenü die Option **Neu»Variable** aus.

Umgebungsvariablen müssen zu Projektbibliotheken gehören. Wenn Sie eine Umgebungsvariable von einem System oder Ordner aus erstellen, das bzw. der nicht zu einer Projektbibliothek gehört, wird eine neue Projektbibliothek für die Variable erstellt.

Mit Hilfe der Umgebungsvariablen-Refnum und der Variablen-Eigenschaften lassen sich Umgebungsvariablen programmatisch in einem Projekt konfigurieren. Zum Erstellen einer Umgebungsvariablen-Refnum muss ein Anzeigeelement von der Eigenschaft “Variablen-Referenz” aus erstellt werden. Mit dieser Refnum können jedoch keine Änderungen an Umgebungsvariablen vorgenommen werden, die nur in einem Prozess verwendet werden.

Senden und Empfangen von Werten einer Umgebungsvariable auf dem Frontpanel

Mit Hilfe der Frontpanel-Datenbindung lassen sich Werte eines Frontpanel-Objekts mit anderen VIs austauschen.



Hinweis Frontpanel-Objekte können nur an Umgebungsvariablen gebunden werden, die über das Netzwerk zugänglich sind.

Ziehen Sie eine Umgebungsvariable aus dem Projekt-Explorer auf das Frontpanel eines VIs, um ein Bedienelement zu erstellen, das an eine Umgebungsvariable gebunden ist. Sie können ein Bedienelement auch mit der rechten Maustaste anklicken, aus dem Kontextmenü die Option **Eigenschaften** auswählen und ein Bedienelement anhand der Optionen auf der Seite **Datenbindung** an eine Umgebungsvariable bzw. an ein Datenobjekt des NI-Protokolls zum Senden und Empfangen (NI-PSP) binden.



Hinweis In LabVIEW 8.0 können Sie nicht mit der rechten Maustaste auf ein Bedienelement klicken und aus dem Kontextmenü **Datenoperation»DataSocket-Verbindung** auswählen. Wählen Sie stattdessen im Dialogfeld **Eigenschaften** des Bedienelements aus dem Pulldown-Menü **Datenbindungsauswahl** die Option **DataSocket** aus, um Frontpanel-Objekte über das `dstp-`, `opc-`, `ftp-`, `http-` oder `file-`Protokoll miteinander zu verbinden.

Wenn die Datenbindung für ein Bedienelement deaktiviert wird, ändert sich der Wert der Umgebungsvariable, an die das Bedienelement gebunden ist, wenn der Wert des Bedienelements geändert wird.

Senden und Empfangen von Werten einer Umgebungsvariable auf dem Blockdiagramm

Ein “Umgebungsvariablen”-Knoten ist ein Blockdiagrammobjekt, das auf die zugehörige Umgebungsvariable im Projekt-Explorer verweist. Mit Hilfe eines Umgebungsvariablenknotens können Werte einer Umgebungsvariable ausgetauscht bzw. der Zeitstempel für eine Umgebungsvariable gelesen werden. Zum Erstellen eines Umgebungsvariablenknotens ziehen Sie eine Umgebungsvariable aus dem Projekt-Explorer in das Blockdiagramm eines VIs im selben Projekt.

Es ist aber auch möglich, einen Umgebungsvariablenknoten in der **Funktionen**-Palette auszuwählen und auf dem Blockdiagramm einzufügen. Zum Binden eines Umgebungsvariablenknotens auf dem Blockdiagramm an eine Umgebungsvariable im aktiven Projekt klicken Sie doppelt auf den Knoten. Daraufhin wird das Dialogfeld **Variable auswählen** angezeigt. Sie können aber auch den Umgebungsvariablenknoten mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Variable auswählen** auswählen.

Beschränken von Umgebungsvariablen auf eine Datenquelle

Per Voreinstellung können mehrere Anwendungen Daten an eine Umgebungsvariable übermitteln. Eine Umgebungsvariable, die über das Netzwerk arbeitet, kann jedoch auch so konfiguriert werden, dass sie nur Wertänderungen von einem System bzw. einer Anwendung annimmt. Aktivieren Sie die Option **Einzelne Datenquelle** auf der Seite **Variable** im Dialogfeld **Eigenschaften für Umgebungsvariable**. Dadurch wird sichergestellt, dass keine andere Anwendung als die angegebene Daten an die Umgebungsvariable überträgt. Die Engine für Umgebungsvariablen beschränkt das Schreibrecht auf ein einziges VI auf einem einzigen Rechner. Nur die erste Anwendung, die eine Verbindung zur Umgebungsvariable aufbaut, kann Werte senden. Wenn die erste Verbindung mit einer

Datenquelle abbricht, kann die zweite Anwendung in der Warteschlange Werte an die Umgebungsvariable übertragen. Anwendungen, die keine Berechtigung zum Senden an die Umgebungsvariable haben, werden entsprechend benachrichtigt.

Aktivieren des Puffers für Umgebungsvariablen

Wenn Sie zum programmatischen Datenaustausch Umgebungsvariablen oder eine psp- oder dstp-URL verwenden, werden per Voreinstellung nur die neuesten Werte an die empfangenden Anwendungen übermittelt. Wenn jedoch Daten schneller gesendet werden, als sie vom anderen Client empfangen werden können, werden ältere, noch nicht verarbeitete Daten von den neuen überschrieben, bevor sie ausgelesen werden können. Wenn der Empfänger einen Wert nicht erhält, bevor der nächste Wert empfangen wird, geht dieser verloren. Ein solcher Datenverlust kann sowohl beim Server als auch beim Client auftreten. Wenn der Empfänger immer nur die neuesten an den Server übertragenen Werte erhalten soll, stellt dieser Datenverlust jedoch normalerweise kein Problem dar. Um jedoch alle Werte der Umgebungsvariable, die an den Server übermittelt wurden, zu empfangen oder wenn der Verlust von Daten nicht akzeptabel ist, müssen die Daten auf dem Client-Rechner in einem Puffer zwischengespeichert werden. Aktivieren Sie dazu die Option **Pufferung verwenden** auf der Seite **Variable** im Dialogfeld **Eigenschaften für Umgebungsvariable**.

Verwalten von Umgebungsvariablen

Wählen Sie **Werkzeuge»Umgebungsvariable»Variablen-Manager**, um zum Dialogfeld **Variablen-Manager** zu gelangen. Hier lassen sich Umgebungsvariablen außerhalb einer Projektumgebung bearbeiten, erstellen und überwachen. Außerdem können Sie mit Hilfe dieses Dialogfelds angeben, wie lange die Engine für Umgebungsvariablen existieren soll.

Erstellen von Quellcodepaketen

Die Optionen für die Weitergabe von VIs, zum Beispiel zum Entfernen des Blockdiagramms oder Festlegen von Passwörtern, die es bisher im Dialogfeld **Mit Optionen speichern** gab, sind jetzt auch für das Erstellen von Quellcodepaketen verfügbar.

Mit Hilfe der **Build-Spezifikationen** im Projekt-Explorer können Build-Spezifikationen für Quellcodepakete erstellt werden. Klicken Sie mit der rechten Maustaste auf **Build-Spezifikationen** und wählen Sie aus dem Kontextmenü **Neu»Quellcodepaket** aus, um zum Dialogfeld **Einstellungen zum Quellcodepaket** zu gelangen.

Neuerungen am Application Builder

Der Application Builder ist im Projekt-Explorer integriert. Wenn Sie mit dem LabVIEW Base Package oder Full Development System arbeiten, müssen Sie den Application Builder separat erwerben. Besuchen Sie dazu die National-Instruments-Website ni.com und geben Sie den Infocode `rdlv21` ein.

Mit den **Build-Spezifikationen** im Projekt-Explorer können Sie Build-Spezifikationen für Anwendungen (EXE-Dateien), DLLs bzw. Shared Libraries und Zip-Dateien vornehmen und anhand dieser Einstellungen Applikationen, DLLs oder Zip-Dateien erzeugen lassen. **(Windows)** Mit den **Build-Spezifikationen** können auch Build-Spezifikationen für Installer angelegt und anhand dieser Einstellungen Installer erstellt werden.

Build-Spezifikationen entsprechen den `*.bld`-Dateien in älteren Versionen des Application Builders. Sie gehören nun jedoch zu einem LabVIEW-Projekt und sind keine eigenständigen Dateien mehr.



Hinweis Wenn Sie die Funktionen des Application Builders verwenden möchten, muss ein Projekt erstellt werden. Dazu ist mit dem **Assistenten zur Erstellung von Anwendungen** von einem geöffneten VI aus ein Projekt für eine Anwendung zu erstellen. Wählen Sie **Werkzeuge»Applikation erzeugen**, um den **Assistenten zur Erstellung von Anwendungen** zu öffnen. Sie werden darin mit Dialogfeldern durch den gesamten Prozess der Erstellung einer Anwendung geführt.

Die Umwandlung einer `*.bld`-Datei in eine Build-Spezifikation ist in einem neuen Projekt möglich.

In LabVIEW 8.0 gibt es folgende Verbesserungen und Änderungen am Application Builder:

- Zu jeder Art von Build-Spezifikation gibt es ein mehrseitiges Dialogfeld, mit dessen Hilfe Sie die Einstellungen für die Build-Spezifikation konfigurieren und bearbeiten können. Klicken Sie mit der rechten Maustaste auf die Build-Spezifikation und wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus, um zu diesem Dialogfeld zu gelangen.
- **(Windows)** Die Build-Spezifikationen für Installationsprogramme sind separat festzulegen. Sie können auch mehrere Builds für Anwendungen oder DLLs in einem Installer-Build einfügen. Klicken Sie mit der rechten Maustaste auf die Build-Spezifikation für das Installationsprogramm, wählen Sie **Eigenschaften** und wählen Sie dann auf der Seite **Quelldateien** die Builds der Anwendungen, DLLs oder Quellcodepakete aus, die im Installationsprogramm enthalten sein sollen.

- Anwendungen und DLLs können auf Fehler untersucht werden, wenn Sie die Fehlersuche in der Build-Spezifikation aktivieren, bevor Sie die Anwendung oder DLL erstellen. Klicken Sie mit der rechten Maustaste auf die Build-Spezifikation, wählen Sie **Eigenschaften** aus dem Kontextmenü aus und aktivieren Sie dann auf der Seite **Fortgeschritten** die Option **Fehlersuche aktivieren**.



Hinweis Wenn Sie Anwendungen und DLLs auf Fehler untersuchen, können allerdings ablaufinvariante Frontpanel, die von der Funktion “VI-Referenz öffnen” erstellt werden, nicht untersucht werden. Ablaufinvariante Frontpanel, die als Eintrittspunkte für von LabVIEW erstellte DLLs dienen, können ebenfalls nicht auf Fehler untersucht werden.

- **(Windows)** Sie können in den Build-Spezifikationen einer Anwendung oder DLL eine Version angeben. Klicken Sie dazu mit der rechten Maustaste auf die Build-Spezifikationen für eine Anwendung, wählen Sie **Eigenschaften** und legen Sie dann auf der Seite **Angaben zur Anwendung** eine Version fest. Klicken Sie mit der rechten Maustaste auf die Build-Spezifikationen für eine DLL, wählen Sie **Eigenschaften** und legen Sie dann auf der Seite **Angaben zur DLL** eine Versionsnummer für die DLL fest.
- **(Windows)** Installer und Treiber für Produkte von National Instruments können ebenfalls in Installer-Builds eingefügt werden. Klicken Sie dazu mit der rechten Maustaste auf die Build-Spezifikation des Installers, wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus und fügen Sie auf der Seite **Zusätzliche Installer** die Treiber, die Unterstützung für die LabVIEW-Runtime-Engine und andere Installer von National Instruments zum Installer-Build hinzu.
- In den Build-Spezifikationen können Sie die Sprache des Laufzeitsystems für eine Anwendung oder DLL festlegen. Klicken Sie dazu mit der rechten Maustaste auf die Build-Spezifikationen für eine Anwendung oder DLL, wählen Sie **Eigenschaften** und legen Sie dann auf der Seite **Sprachen des Laufzeitsystems** die Spracheinstellungen für die Anwendung bzw. DLL fest.
- Die LabVIEW Runtime-Engine ist mehrsprachig. Sie müssen also nicht mehrere Versionen davon in eine Anwendung, DLL oder einen Installer einfügen.
- **(Windows)** Sie können Build-Spezifikationen für Installer so konfigurieren, dass Registry-Einträge eingefügt und mehrere Verknüpfungen zu Dateien im Installer erstellt werden können. Klicken Sie dazu mit der rechten Maustaste auf die Build-Spezifikationen des Installers, wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus und erstellen bzw. konfigurieren Sie auf der Seite **Registry** die Registry-Schlüssel. Auf der Seite **Verknüpfungen** lassen sich Verknüpfungen erstellen und konfigurieren.

- Wenn Sie ein Installationsprogramm für eine Anwendung oder DLL erstellen, in denen die VIs zur Datenspeicherung verwendet werden, müssen Sie den NI-USI-Installer hinzufügen. Um einen NI-USI-Installer aufzunehmen, klicken Sie mit der rechten Maustaste auf die Build-Spezifikationen des Installers, wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus und klicken Sie auf der Seite **Zusätzliche Installer** unter **Einzubeziehende Installer von National Instruments** auf **NI USI**.

Neue Palettenaufteilung

Folgende Verbesserungen wurden in LabVIEW 8.0 an den Paletten **Elemente** und **Funktionen** vorgenommen:

- Die Palettenobjekte sind nach Kategorien geordnet. Die Formate **Standard** und **Standard (Symbol und Text)** wurden durch **Kategorie (Standard)** und **Kategorie (Symbol und Text)** ersetzt. In der folgenden Tabelle sehen Sie die Hauptpaletten der **Elemente**-Palette aus LabVIEW 7.1 und wo sich die entsprechende Kategorie in LabVIEW 8.0 befindet:

Elementepalette

Tabelle 2. Aufbau der Elemente-Palette

Palette in 7.1	Kategorie in 8.0
Express	Express
Numerisch	Modern
Boolesch	Modern
String & Pfad	Modern
Array & Cluster	Modern
Liste & Tabelle	Modern
Graph	Modern
Ring & Enum	Modern
Container	Modern
I/O	Modern
Refnum	Modern
Gestaltungselemente	Modern
Dialogelemente	System
Klassische Elemente	Klassisch

Funktionenpalette

Tabelle 3. Aufbau der Funktionen-Palette

Palette in 7.1	Kategorie in 8.0
Express	Express
Strukturen	Programmierung
Numerisch	Programmierung
Boolesch	Programmierung
String	Programmierung
Array	Programmierung
Cluster	Programmierung
Vergleich	Programmierung
Zeit & Dialog	Programmierung
Datei-I/O	Programmierung
NI-Messtechnik	Mess-I/O
Signalverlauf	Programmierung
Analyse	Mathematik und Signalverarbeitung
Instrumenten-I/O	Instrumenten-I/O
Anwendungssteuerung	Programmierung
Audio & Grafik	Programmierung
Kommunikation	Konnektivität und Datenkommunikation
Erstellen von Reports	Programmierung
Fortgeschritten	Konnektivität und Datenkommunikation
Gestaltungselemente	Programmierung

- In den Anzeigeformaten **Kategorie (Standard)** und **Kategorie (Symbol und Text)** können Sie die Reihenfolge der Kategorien ändern, indem Sie eine Kategorie mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Kategorie nach oben verschieben** oder **Kategorie nach unten verschieben** auswählen. Sie können aber auch den doppelten Strich auf der linken Seite der Kategorie anklicken und den Pfeil nach oben oder unten verschieben.

- LabVIEW wird per Voreinstellung mit einer gekürzten Palettenansicht installiert, es sei denn, Sie hatten vorher eine andere LabVIEW-Version, in der Sie die Palettenansicht **Fortgeschritten** verwendet haben. In der gekürzten Palettenansicht sind einige erweiterte Kategorien der Hauptpaletten ausgeblendet. Zum Einblenden aller Kategorien in der Palette **Elemente** bzw. **Funktionen** klicken Sie in der Symbolleiste auf die Schaltfläche **Ansicht** und wählen Sie aus dem Kontextmenü **Ständig sichtbare Kategorien»Alle Kategorien anzeigen** aus.
- Sie können entweder das Format der aktuellen Palette oder aller Paletten ändern. Klicken Sie auf die Schaltfläche **Ansicht** in der Symbolleiste der **Elemente**- oder **Funktionen**-Palette und wählen Sie aus dem Kontextmenü **Diese Palette anzeigen als** ein Format aus, um das Format der aktuellen Palette zu ändern. Um ein Format für alle Paletten auszuwählen, klicken Sie auf **Werkzeuge»Optionen** und öffnen Sie die Seite “Paletten ‘Elemente’ und ‘Funktionen’”.
- Beim **Baumstruktur-Format** werden die Palettenobjekte in Form einer Baumstruktur angezeigt. Damit die aktuelle Palette als Baumstruktur angezeigt wird, klicken Sie in der Symbolleiste der **Elemente**- oder **Funktionen**-Palette auf die Schaltfläche **Ansicht** und wählen Sie **Diese Palette anzeigen als»Baumstruktur**.
- In den Formaten **Text** und **Baumstruktur** können Sie alle Elemente derselben Ebene alphabetisch anzeigen, indem Sie auf die Schaltfläche **Ansicht** klicken und aus dem Kontextmenü **Alphabetisch sortieren** auswählen.
- In der Kategorie **Favoriten** können Sie häufig genutzte Elemente der **Funktionen**-Palette zusammenfassen. Sie können der Kategorie **Favoriten** in den Anzeigeformaten **Kategorie (Standard)**, **Kategorie (Symbol und Text)**, **Symbole** und **Symbole und Text** neue Elemente hinzufügen. Klicken Sie mit der rechten Maustaste auf ein Element in der fixierten **Funktionen**-Palette und wählen Sie aus dem Kontextmenü die Option **Objekt zu Favoriten hinzufügen** aus, um der Kategorie **Favoriten** ein Objekt hinzuzufügen. In den Formaten **Kategorie (Standard)** und **Kategorie (Symbole und Text)** können Sie eine Palette auch erweitern, um eine Unterpalette anzuzeigen. Klicken Sie dazu mit der rechten Maustaste auf den Titel der Unterpalette und wählen Sie aus dem Kontextmenü die Option **Unterpalette zu Favoriten hinzufügen** aus.

Verschobene und umbenannte Paletten

Die folgenden Paletten wurden in LabVIEW 8.0 verschoben oder umbenannt:

- Auf der Palette **VISA: Fortgeschritten** wurden die Unterpaletten **Schnittstellenspezifisch** und **High-Level-Registerzugriff** in **Bus-/Schnittstellenspezifisch** bzw. **Registerzugriff** umbenannt.

- Die VIs der Palette **Erweiterte Formelanalyse** befinden sich nun auf den Paletten **Analyse von Formeln** bzw. **1D- & 2D-Entwicklung**.
- Die VIs der Paletten **Cluster** und **Variant** befinden sich nun auf der Palette **Cluster & Variant**.
- Die Paletten **Arrayoperationen**, **Kurvenanpassung**, **Formeln**, **Logarithmisch** und **Spezielle und numerische Funktionen** wurden in **Signaloperation**, **Anpassung**, **Skripte und Formeln**, **Exponentialfunktionen** bzw. **Grund- und Spezialfunktionen** umbenannt. Einige der Kurvenanpassungs-VIs sind nun auf der Palette **Interp & Extrap** zu finden.
- Die Palette **Numerische Konstanten (Express)** wurde in **Mathematische und wissenschaftliche Konstanten (Express)** umbenannt.
- Die Palette **Weitere numerische Konstanten** wurde in **Mathematische und wissenschaftliche Konstanten** umbenannt.
- Die Palette **Logarithmisch (Express)** wurde in **Logarithmische Funktionen (Express)** umbenannt.
- Die VIs der Palette **Arrayoperationen (Punkt für Punkt)** befinden sich nun auf den Paletten **Geometrie (Punkt für Punkt)**, **Polynom (Punkt für Punkt)**, **Wahrscheinlichkeit & Statistik (Punkt für Punkt)** und **Signaloperation (Punkt für Punkt)**.
- Die VIs der Palette **Kurvenanpassung (Punkt für Punkt)** sind nun auf den Paletten **Anpassung (Punkt für Punkt)** und **Interpolation (Punkt für Punkt)** zu finden.
- Die Palette **Fortgeschritten** wurde in **Bibliotheken & Programme** umbenannt.
- Die VIs der Palette **Zeit & Dialog** sind nun auf den Paletten **Timing** und **Dialog & Benutzeroberfläche** zu finden.
- Die VIs der Palette **Frequenzbereich** befinden sich nun auf den Paletten **Spektrumanalyse** und **Transformationen**.
- Die VIs der Palette **Frequenzbereich (Punkt für Punkt)** sind nun auf den Paletten **Spektrumanalyse (Punkt für Punkt)** und **Transformationen (Punkt für Punkt)** zu finden.
- Die VIs der Palette **Zeitbereich** befinden sich nun auf den Paletten **Signaloperation** und **Integ & Diff**.
- Die VIs der Palette **Zeitbereich (Punkt für Punkt)** befinden sich nun auf den Paletten **Signaloperation (Punkt für Punkt)** und **Integral & Differential (Punkt für Punkt)**.
- Die Palette **Zeitgesteuerte Schleife** wurde in **Zeitgesteuerte Strukturen** umbenannt.
- Die Palette **Alle VIs und Funktionen** ist nicht mehr vorhanden.

Verbesserungen bei der Palettenbearbeitung

In LabVIEW 8.0 gab es folgende Neuerungen beim Bearbeiten von Paletten:

- In LabVIEW 7.x und Vorgängerversionen können Sie eine individuelle Ansicht der Paletten **Elemente** und **Funktionen** einrichten. In LabVIEW 8.0 gibt es keine benutzerdefinierten Palettenansichten mehr. Sie können jetzt eine Palette bearbeiten, ohne eine benutzerdefinierte Palettenansicht zu verwenden. Wenn Sie mit verschiedenen Zielsystemen arbeiten, wird für jedes System eine Palette geladen. Zum Bearbeiten einer Palette für ein bestimmtes System öffnen Sie von einem VI, das auf dem System läuft, das Dialogfeld **Elemente- und Funktionen-Palette bearbeiten**. Wählen Sie **Werkzeuge»Fortgeschritten»Palette bearbeiten**, um zum Dialogfeld **Elemente- und Funktionen-Palette bearbeiten** zu gelangen.
- Alle Änderungen an einer Palette werden im Ordner 8.0\Palettes im Standarddatenverzeichnis gespeichert.
- Um ein Objekt auszublenden, das es im Verzeichnis gibt, in dem der Inhalt einer Palette gespeichert ist, klicken Sie mit der rechten Maustaste auf das Element und wählen Sie **Synchronisiertes Element ausblenden** aus dem Kontextmenü aus.
- Ist ein Element mit einem Verzeichnis synchronisiert, können Sie sich den Pfad zum Quellverzeichnis anzeigen lassen, indem Sie dieses Element mit der rechten Maustaste anklicken und aus dem Kontextmenü **Synchronisierungspfad anzeigen** auswählen.
- Wenn eine Zeile oder Spalte am Ende einer Palette angezeigt wird und nur synchronisierte Elemente enthält, die ausgeblendet sind bzw. Elemente, die LabVIEW nicht finden kann, dann ist die Zeile oder Spalte nur während der Palettenbearbeitung sichtbar.
- Leere Unterpaletten und Unterpalletten, die nur Elemente enthalten, die nicht gefunden werden können, sind nur während der Bearbeitung der Palette sichtbar.

Neuordnung des Menüs

In LabVIEW gibt es neue Menüs und viele Menüpunkte sind verschoben oder umbenannt worden. Folgende Menüpunkte sind im LabVIEW-Menü geändert worden:

- **Datei**—Hinzugefügte Menüpunkte für wichtige Projektinformationen sind zum Beispiel: “Projekt öffnen”, “Projekt speichern” oder “Projekt schließen”.
- **Bearbeiten**—Einige Menüpunkte unter **Ausführen** sind nach **Bearbeiten** verschoben worden und der Menüpunkt **Alle auswählen** ist hinzugekommen.

- **Anzeigen**—Neu hinzugefügtes Menü, in dem Menüpunkte zum Anzeigen von Fenstern und Paletten, Durchsuchen von Klassen und Anzeigen des Fensters **Projekt-Explorer** enthalten sind. Einige **Fenster**-Menüpunkte sind nun unter **Anzeigen** zu finden. Da das Menü **Durchsuchen** entfernt wurde, enthält nun das Menü **Anzeigen** viele Menüpunkte aus **Durchsuchen**. Wählen Sie **Anzeigen»Beziehungen ermitteln**, um nach Menüpunkten des früheren Menüs **Durchsuchen** zu suchen.
- **Projekt**—Neu hinzugefügtes Menü mit Menüpunkten zum Hinzufügen von VIs und Dateien zu einem Projekt, Erstellen von Build-Spezifikationen und zur Anzeige von Projektinformationen.
- **Ausführen**—Beim Menü **Ausführen** sind einige Menüpunkte zur Fehlersuche in VIs hinzugefügt bzw. entfernt worden. Da das Menü **Durchsuchen** entfernt wurde, enthält nun das Menü **Ausführen** viele Menüpunkte aus **Durchsuchen**.
- **Werkzeuge**—Dem Menü **Werkzeuge** sind verschiedene Menüpunkte hinzugefügt worden. Viele der vorhandenen Menüpunkte aus **Werkzeuge** wurden in Untermenüs unterteilt.
- **Fenster**—Menüpunkte zur Palettenansicht sowie die Fenster **Naviga-tion** und **Fehlerliste** sind entfernt worden. Diese Menüpunkte sind nun unter dem Menü **Anzeigen** zu finden.
- **Hilfe**—Es wurden Menüpunkte zum Suchen von Instrumententreibern und zum Aktivieren von LabVIEW hinzugefügt.



Hinweis Einige Menüpunkte sind jedoch nur unter bestimmten Betriebssystemen zu sehen, mit bestimmten LabVIEW-Entwicklungsumgebungen oder wenn im Projekt-Explorer ein Objekt oder VI ausgewählt ist.

Weitere Informationen zu neuen Namen und wo sich bestimmte Menüs befinden, sind in der *LabVIEW-Hilfe* zu finden.

Verwenden der Versionsverwaltung

Wenn Sie auf Ihrem Rechner einen Versionsverwaltungs-Provider von einem Drittanbieter installiert haben, wie zum Beispiel Perforce oder Microsoft Visual SourceSafe, können Sie in einem LabVIEW-Projekt oder mit einzelnen VIs, die nicht zu einem Projekt gehören, die Versionsverwaltung verwenden. Operationen der Versionsverwaltung lassen sich an VIs, Dateien innerhalb eines Projekts und Projektdateien, wie Projektbibliotheken, vornehmen. LabVIEW ermöglicht die Zusammenarbeit mit einem Versionsverwaltungs-Provider eines Drittanbieters. Zum Verwenden der Versionsverwaltung in LabVIEW muss zuerst ein Versionsverwaltungs-Provider eines Drittanbieters auf dem Rechner installiert werden.

(Windows) LabVIEW arbeitet mit jedem Versionsverwaltungs-Provider zusammen, der das Microsoft Source Code Control Interface unterstützt.
(Mac OS und Linux) Es kann das Perforce-Kommandozeilenfeld verwendet werden.

Im Hinblick auf den Funktionsumfang der Versionsverwaltung gibt es bei LabVIEW 8.0 die nachfolgenden Änderungen.



Hinweis Versionsverwaltung ist nur mit dem LabVIEW-Professional Development System möglich.

- Den in LabVIEW integrierten Versionsverwaltungs-Provider von LabVIEW 7.x und Vorgängerversionen gibt es in LabVIEW 8.0 nicht mehr. Weitere Informationen zur Umstellung von Versionsverwaltungsdateien aus früheren LabVIEW-Versionen finden Sie im Abschnitt *Umstellung vom integrierten LabVIEW-Versionsverwaltungs-Provider*.
- Nachdem Sie einen Versionsverwaltungs-Provider eines Drittanbieters ausgesucht und installiert haben, muss LabVIEW für die Zusammenarbeit mit diesem Provider konfiguriert werden.
- Mit der Versionsverwaltung in LabVIEW können Sie Dateien zum Bearbeiten auschecken, bearbeitete Dateien einchecken, die neueste Version einer Datei vom Versionsverwaltungsprogramm abrufen und eine Datei mit der neuesten Version im Versionsverwaltungsprogramm vergleichen.
- Sie können sich die Historie der Versionsverwaltung einer Datei anzeigen lassen.
- Mit Hilfe der Versionsverwaltungs-VIs können Sie Operationen der Versionsverwaltung programmatisch durchführen.
- In LabVIEW können keine Perforce-Konfigurationsdateien erstellt werden.

Umstellung vom integrierten LabVIEW-Versionsverwaltungs-Provider

Den in LabVIEW integrierten Versionsverwaltungs-Provider von LabVIEW 7.x und Vorgängerversionen gibt es in LabVIEW 8.0 nicht mehr. Wenn Sie in LabVIEW die Versionsverwaltung nutzen möchten, müssen Sie einen Versionsverwaltungs-Provider von einem Drittanbieter auswählen. Wenn Sie bisher in früheren LabVIEW-Versionen den integrierten Provider verwendet haben, müssen Sie die Dateien auf einen anderen Versionsverwaltungs-Provider in LabVIEW umstellen. Die aktuelle Liste der Versionsverwaltungs-Provider von Drittanbietern, mit denen LabVIEW arbeitet, finden Sie auf der Website von National Instruments unter ni.com nach Eingabe des Infocodes `rdbp01`.

Beim Umstellen von Dateien auf einen neuen Versionsverwaltungs-Provider geht die Versionshistorie verloren, die im integrierten Provider gespeichert war. Ältere Versionen der Dateien lassen sich nicht auf den neuen Provider übertragen.

Führen Sie folgende Schritte aus, um Dateien vom integrierten Versionsverwaltungs-Provider auf einen Versionsverwaltungs-Provider von einem Drittanbieter zu übertragen:

1. Vergewissern Sie sich, dass die Dateien im integrierten Versionsverwaltungs-Provider früherer LabVIEW-Versionen von allen Nutzern eingchecked sind.
2. Rufen Sie auf dem Computer, auf dem Sie den Versionsverwaltungs-Provider umstellen möchten, im integrierten Provider die neueste Version aller Dateien ab.
3. Checken Sie im integrierten Provider die betreffenden Dateien aus der Versionsverwaltung aus.
4. Nehmen Sie im Versionsverwaltungs-Provider des Drittanbieters die im neuen Versionsverwaltungsprojekt gewünschten Einstellungen vor.
5. Konfigurieren Sie LabVIEW für die Zusammenarbeit mit dem neuen Versionsverwaltungs-Provider.
6. Erstellen Sie ein LabVIEW-Projekt. Fügen Sie dem Projekt die Dateien hinzu, die sich im integrierten Versionsverwaltungs-Provider befinden. Fügen Sie die Dateien der Versionsverwaltung hinzu, wenn Sie dazu aufgefordert werden. Sie können die Dateien auch direkt in den Provider des Drittanbieters einfügen.

Verbesserungen an VIs und Funktionen

Nachfolgend sind alle Verbesserungen und Änderungen an den VIs und Funktionen in LabVIEW 8.0 beschrieben.

Neuerungen bei den Analyse-VIs

Die VIs der Palette **Analyse** in LabVIEW 7.x und Vorgängerversionen befinden sich nun auf den Paletten **Mathematik**, **Signalverarbeitung** und **Programmierung**. Die Symbole für viele dieser VIs haben sich geändert. Das gelbe Banner im oberen Teil des Symbols zeigt die Unterpalette an, auf der sich das VI befindet.

Verbesserungen und Veränderungen an VIs

Die folgenden Mathematik-, Signalverarbeitungs- und Programmierungs-VIs wurden in LabVIEW 8.0 verbessert bzw. verändert:

Palette “Mathematik”

Die VIs zur Anpassung wurden folgendermaßen verbessert und verändert:

- Das Anpassungsverfahren wird nun am Eingang **Methode** der VIs “Exponentialanpassung”, “Koeffizienten der Exponentialanpassung”, “Lineare Anpassung” und “Koeffizienten der Linearanpassung” festgelegt. Es gibt die Anpassungsverfahren “Kleinste Quadrate”, “Kleinster Restbetrag” und “Biquadrat”. Diese VIs haben auch einen **Toleranz**-Eingang, an dem festgelegt wird, wann die iterative Anpassung der Ausgangsparameter beendet werden soll (sofern beim ausgewählten Verfahren Iterationen erforderlich sind). Am Eingang **Gewichtung** der VIs wird die Gewichtung der beobachteten Datenwerte festgelegt.
- Der Ausgang **Restbetrag** der VIs “Exponentialanpassung” und “Lineare Anpassung” gibt den gewichteten mittleren Fehler des Anpassungsmodells aus.
- Am Eingang **verfeinern?** der VIs “Exponentialanpassung” und “Koeffizienten der Exponentialanpassung” wird festgelegt, ob die **Amplitude** und die **Dämpfung** des Anpassungsmodells weiter verfeinert werden sollen.
- Beim VI “Exponentialanpassung” gibt es den Eingang **Standardabweichung** und den Ausgang **mse** nicht mehr.
- Das VI “Linearanpassung” hat keinen **mse**-Ausgang mehr, wie bei LabVIEW 7.x und Vorgängerversionen.
- Der Eingang **Gewichtung** des VIs “Allgemeine LS-Anpassung” legt die Gewichtung der Beobachtungswerte fest. Dieses VI hat auch einen neuen Algorithmus, **SVD für rangdefiziente Matrix H**, um die beste Anpassung zu berechnen. Verwenden Sie diesen Algorithmus nur dann, wenn die Eingangsmatrix rangdefizient ist und kein anderer Algorithmus erfolgreich war. Das VI hat keinen Eingang **Standardabweichung** wie in LabVIEW 7.x und Vorgängerversionen.
- Am Eingang **Koeffizientenbedingung** des VIs “Allgemeine Polynom-Anpassung” werden die Einschränkungen bei Polynom-Koeffizienten bestimmter Ordnungen festgelegt. Der Ausgang **Polynom-Anpassungs-Koeffizienten** des VIs wurde in **Polynom-Koeffizienten** umbenannt.

Die VIs zur Integral- und Differentialrechnung wurden folgendermaßen verbessert und verändert:

- Mit dem VI “Numerische Integration” können zwei- und dreidimensionale numerische Integrationen durchgeführt werden.

Die VIs für Lineare Algebra wurden folgendermaßen verbessert und verändert:

- In LabVIEW 7.x ist der Ausgang **Eigenvektor** des VIs “Eigenwerte und Vektoren” normalisiert, so dass die größte Komponente stets den Wert 1 hat. In LabVIEW 8.0 ist der Ausgang **Eigenvektor** normalisiert, so dass dessen euklidische Norm 1 lautet.
- Das VI “QR-Zerlegung” ersetzt das VI “QR-Faktorisierung”. Das VI “QR-Zerlegung” hat zwei neue Eingänge. Mit **Pivotieren?** wird festgelegt, ob die Eingangsmatrix **A** mit Spaltenpivotisierung zerlegt werden soll. **Q-Option** gibt an, wie das VI die orthogonale Matrix **Q** erzeugt. Dieses VI kann eine Matrix maximaler bzw. ökonomischer Größe aber auch gar keine orthogonale Matrix erzeugen. Der Ausgang **P** dieses VIs gibt die Permutationsmatrix aus.
- Das VI “QZ-Zerlegung” hat zwei neue Eingänge. Mit **Zerlegungstyp** wird festgelegt, ob eine allgemeine Hessenberg- oder eine allgemeine Schur-Zerlegung durchgeführt werden soll, und mit **Ordnung** wird bestimmt, wie die verallgemeinerten Eigenwerte **Alpha** und **Beta** angeordnet werden sollen.
- Der Eingang **Reihenfolge** des VIs “Schur-Zerlegung” legt die Reihenfolge der **Eigenwerte** entsprechend der **Schur-Form** und **Schur-Vektoren** fest.

Die Polynom-VIs wurden folgendermaßen verbessert und verändert:

- Mit dem VI “Nullstellen eines Polynoms” lassen sich die Nullstellen eines komplexen Polynoms ermitteln. Außerdem enthält das VI die Option **Erweiterte Verfeinerung** zur Eingrenzung der Suche.
- Am Ausgang **Anzahl der Nullkoeffizienten** des VIs “Null-Koeffizienten entfernen” wird die Anzahl der abschließenden Koeffizienten angegeben, die sich um 0 bewegen und von LabVIEW entfernt worden sind.

Die VIs für Wahrscheinlichkeit und Statistik wurden folgendermaßen verbessert und verändert:

- Das VI “Modalwert” ist nun polymorph und Sie können damit nach einem oder mehreren Modalwerten suchen. Am Eingang **Intervalle** dieses VIs können auch nicht positive Werte anliegen. Wenn der Eingang **Intervalle** nicht positiv ist, berechnet das VI nur die exakten Modalwerte der Eingangswerte.

- Am **X**-Eingang des VIs “Standardabweichung und Varianz” wird nun empfohlen, einen Wert anzugeben. Es ist aber nicht unbedingt erforderlich.

Die VIs für **Rationale Polynome** wurden folgendermaßen verbessert und verändert:

- Der Eingang **c[0..n+m]** des VIs “Pade-Approximation” wurde in **Ableitungen** umbenannt. Der Ausgang **A[0..m]** wurde in **Zähler** und das Anzeigeelement **B[0..n]** in **Nenner** umbenannt.

Palette “Signalverarbeitung”

Die VIs für **Fortgeschrittene FIR-Filterung** wurden folgendermaßen verbessert und verändert:

- Der Eingang **Fensterparameter** des VIs “FIR-gefensterte Koeffizienten” gibt bestimmte Parameterwerte für die Kaiser-, Gauß- und Dolph-Chebyshev-Fenster an. Der Datentyp des Eingangs **Fenster** für dieses VI hat sich geändert. Er lautet nicht mehr “Enum”, sondern “vorzeichenloser 32-Bit-Integer”.

Die VIs für **Fortgeschrittene IIR-Filterung** wurden folgendermaßen verbessert und verändert:

- Das VI “Kaskadierung->Direkte Koeffizienten” wurde in “Kaskadierung in direkte Koeffizienten” umbenannt.
- Die VIs “IIR-Kaskadenfilter”, “IIR-Kaskadenfilter mit I.C.”, “IIR-Filter” und “IIR-Filter mit I.C.” sind polymorph und arbeiten mit komplexen Werten.

Die Filter-VIs wurden folgendermaßen verbessert und verändert:

- Am Eingang **Fensterparameter** des VIs “FIR-gefensteter Filter” werden bestimmte Parameterwerte für das Kaiser-, Gauß- und Dolph-Chebyshev-Fenster angegeben. Der Datentyp des Eingangs **Fenster** für dieses VI hat sich geändert. Er lautet nicht mehr “Enum”, sondern “vorzeichenloser 32-Bit-Integer”.
- Alle VIs der Palette **Filter** mit Ausnahme des VIs “Median-Filter” sind polymorph und arbeiten mit komplexen Werten.

Die VIs zur **Signalerzeugung** wurden folgendermaßen verbessert und verändert:

- Das VI “Binäre MLS” hat jetzt einen **Fehlercode**-Ausgang.
- Am Eingang **Ende ausschließen** des VIs “Verlaufsmuster” wird festgelegt, ob der Endwert im erzeugten Array enthalten sein soll.
- Am Eingang **Signaltyp** des VIs “Signalerzeugung über Zeit” wird nun empfohlen, einen Wert anzugeben. Es ist aber nicht unbedingt erforderlich.

Die VIs zur **Signalerzeugung (Punkt für Punkt)** wurden folgendermaßen verbessert und verändert:

- Das VI “Sägezahnschwingung (Punkt für Punkt)” wurde optimiert und enthält nun eine neue Formel:

$$\mathbf{S\ddot{a}gezahnschwingung} = \text{Amplitude} \times \text{S\ddot{a}gezahn}(q)$$

wobei

$$\text{S\ddot{a}gezahn}(q) = \begin{cases} \frac{q}{180} & \text{wenn } q < 180 \\ \frac{q}{180} - 2 & \text{in allen anderen F\ddot{a}llen} \end{cases}$$

und

$$q = (360 \cdot \mathbf{f} \cdot \mathbf{Zeit}) + \mathbf{Phase} \bmod (360), \text{ Phase in Grad.}$$

Die VIs für Signaloperationen wurden folgendermaßen verbessert und verändert:

- Die VIs “Autokorrelation”, “Faltung” und “Kreuzkorrelation” sind polymorph und arbeiten nun mit 2D-Arrays aus reellen und komplexen Eingangswerten.
- Am Eingang **Normalisierung** der 1D-Instanz der VIs “Autokorrelation” und “Kreuzkorrelation” wird die Normalisierungsmethode bestimmt, mit der die Autokorrelation bzw. Kreuzkorrelation berechnet wird.
- Am Eingang **Ausgabekapazität** der 2D-Instanz des VIs “Faltung” wird die Größe der Faltung festgelegt.
- Die VIs “Vektor normalisieren” und “Matrix normalisieren” wurden im polymorphen VI “Normalisieren” zusammengefasst.
- Die VIs “Schnellskalierung 1D” und “Schnellskalierung 2D” sind jetzt im polymorphen VI “Schnellskalierung” zusammengefasst. Der Parameter $\mathbf{Y_{ij} = X_{ij} / \text{Max}\{X\}}$ der Instanz “Schnellskalierung (2D)” des VIs “Schnellskalierung” wurde in $\mathbf{Y_{ij} = X_{ij} / \text{Max}\{X\}}$ umbenannt.
- Die VIs “Neuaufnahme (Konstante nach Konstante)” und “Neuaufnahme (Konstante nach Variable)” arbeiten nun auch mit komplexen Werten.
- Die VIs “Skalierung 1D” und “Skalierung 2D” wurden im polymorphen VI “Skalieren” zusammengefasst.
- Das VI “Einheitsvektor” ist nun polymorph und hat zwei neue Eingänge. Mit **Norm-Typ** wird der Norm-Typ angegeben und mit **Benutzerdef. Norm** wird der Norm-Typ angegeben, wenn dafür **benutzerdefiniert** ausgewählt wurde.

- Das VI “Mit Nullen auffüllen” ist jetzt polymorph und arbeitet auch mit komplexen Zahlen. Mit dem Eingang **Nur Werte, die keine Potenz von 2 sind?** dieses VIs wird festgelegt, ob die Größe des Eingangsarrays verdoppelt werden soll, wenn die Arraygröße bereits eine gültige Potenz von 2 ist.

Die VIs zur **Spektrumanalyse** wurden folgendermaßen verbessert und verändert:

- Das VI “Leistungsspektrum” ist nun polymorph und arbeitet mit komplexen Zahlen.
- Das VI “STFT-Spektrogramm” wurde in “STFT-Spektrogramme” umbenannt und hat zwei neue Eingänge. Mit **Zeit-Frequenz-Abtastinformation** wird der Zeitschritt, mit dem das Signal im kombinierten Zeit-Frequenz-Bereich abgetastet wird, und die Größe des resultierenden 2D-Zeit-Frequenz-Arrays festgelegt. Unter **Fensterinformation** werden Angaben zum Fenster gemacht, anhand dessen die Kurzzeit-Fouriertransformation (STFT) berechnet werden soll. Es gibt Fenster des Typs “Blackman-Harris”, “Exact-Blackman”, “Flat-Top”, “4 Term B-Harris”, “7 Term B-Harris”, “Niedrige Nebenschwingung”, “Blackman-Nuttall” und “Dreieck”. Die Eingänge **Zeitinkrement**, **Fensterlänge** und **Fensterauswahl** aus LabVIEW 7.x und Vorgängerversionen gibt es nicht mehr.

Die VIs zur Transformation wurden folgendermaßen verbessert und verändert:

- Die VIs “FFT” und “Inverse FFT” arbeiten jetzt mit 2D-Arrays aus reellen und komplexen Werten.
- Der Eingang **Verschiebung?** der VIs “FFT” und “Inverse FFT” legt fest, ob die DC-Komponente in das Zentrum der schnellen Fourier-Transformation verschoben werden soll.
- Am Eingang **FFT-Größe** der 1D-Instanzen des VIs “FFT” wird die Länge der durchzuführenden FFT angegeben.

Die VIs zur Transformation (Punkt für Punkt) wurden folgendermaßen verbessert und verändert:

- Die VIs “Reelle FFT (Punkt für Punkt)” und “Komplexe FFT (Punkt für Punkt)” wurden im polymorphen VI “FFT (Punkt für Punkt)” zusammengefasst.
- Die VIs “Inverse Reelle FFT (Punkt für Punkt)” und “Inverse komplexe FFT (Punkt für Punkt)” wurden im polymorphen VI “Inverse FFT (Punkt für Punkt)” zusammengefasst.

Die VIs zur Signalverlaufskonditionierung enthalten folgende Veränderungen und Verbesserungen:

- Das VI “Skaliertes Fenster” arbeitet nun auch mit komplexen Werten. Am Eingang **Fensterparameter** des VIs werden bestimmte Parameterwerte für das Kaiser-, Gauß- und Dolph-Chebyshev-Fenster angegeben. Der Datentyp des Eingangs **Fenster** für dieses VI hat sich geändert. Er lautet nicht mehr “Enum”, sondern “vorzeichenloser 32-Bit-Integer”.

Die VIs zur Signalverlaufsmessung enthalten folgende Veränderungen und Verbesserungen:

- Der Datentyp des Eingangs **Fenster** hat sich geändert. Für die VIs “Kreuzspektrum (Betrag-Phase)”, “Kreuzspektrum (Real-Im)”, “FFT-Leistungsspektraldichte”, “FFT-Leistungsspektrum”, “FFT-Spektrum (Betrag-Phase)”, “FFT-Spektrum (Real-Im)”, “Übertragungsfunktion (Betrag-Phase)” und “Übertragungsfunktion (Real-Im)” lautet der Datentyp nicht mehr “Enum”, sondern “vorzeichenloser 32-Bit-Integer”.

Die Fenster-VIs wurden folgendermaßen verbessert und verändert:

- Der Eingang **Verhältnis** des VIs “Cosinus-Tapered-Fenster” legt das Verhältnis des zugespitzten Kurvenabschnitts zur Gesamtlänge des Signals fest. Der Ausgang **Cosinus-Tapered {X}** dieses VIs wurde in **Gefensterter X** umbenannt.
- Am **Fensterparameter** des VIs “Skaliertes Zeitbereichsfenster” werden die Parameter für das zu verwendende Fenster festgelegt. Dieses VI hat auch einen neuen **Fehler**-Ausgang. Der Eingang **Signalverlauf** und die Ausgänge **Gefensterter Signal** bzw. **Fensterkonstanten** dieses VIs wurden in **X**, **Gefensterter X** und **Fensterigenschaften** umbenannt. Das VI “Skaliertes Zeitbereichsfenster” hat nun auch ein Fenster des Typs “Blackman-Nuttall”, “Dreieck”, “Kaiser”, “Dolph-Chebyshev” und “Gauß”. Der Datentyp des Eingangs **Fenster** für dieses VI hat sich geändert. Er lautet nicht mehr “Enum”, sondern “vorzeichenloser 32-Bit-Integer”.
- Alle VIs der Palette **Fenster** mit Ausnahme des VIs “Fenster-Eigenschaften” sind jetzt polymorph und arbeiten mit komplexen Zahlen.

Palette “Programmierung”

Die Funktionen zum Vergleich wurden folgendermaßen verbessert und verändert:

- Mit den Vergleichsfunktionen “Gleich?” und “Ungleich?” kann bestimmt werden, ob zwei VI-Server-Referenzen auf dasselbe Objekt verweisen.

Neue VIs und Funktionen

In LabVIEW 8.0 werden folgende Mathematik-, Signalverarbeitungs- und Programmierungs-VIs eingeführt:

Palette “Mathematik”

Auf der Unterpalette **Fortgeschrittene Kurvenanpassung** gibt es folgende neue VIs:

- VI “Intervalle der Exponentialanpassung”
- VI “Koeffizienten für Anpassung der gaußschen Kurve”
- VI “Intervalle der gaußschen Anpassung”
- VI “Anpassung”
- VI “Intervalle der linearen Anpassung”
- VI “Koeffizienten der logarithmischen Anpassung”
- VI “Intervalle der logarithmischen Anpassung”
- VI “Koeffizienten der Leistungsanpassung”
- VI “Intervalle der Leistungsanpassung”
- VI “Abweichende Werte entfernen”

Auf der Unterpalette **Differentialgleichungen** ist das neue VI “ODE-Lösungskalkulator” enthalten.

Auf der Unterpalette **Diskrete Mathematik** gibt es folgende neue VIs:

- VI “Größter gemeinsamer Teiler”
- VI “Kleinstes gemeinsames Vielfaches”
- VI “Vertauschen”

Auf der Unterpalette **Exponentialfunktionen** gibt es die neue Funktion “y-te Wurzel von x”.

Auf der Unterpalette **Anpassung** gibt es folgende neue VIs:

- VI “Anpassung mit kubischer Spline-Interpolation”
- VI “Anpassung der gaußschen Kurve”
- VI “Logarithmische Anpassung”
- VI “Nichtlineare Kurvenanpassung”
- VI “Leistungsanpassung”

Auf der Unterpalette **Geometrie** gibt es folgende neue VIs:

- VI “Verschiebung kartesischer 2D-Koordinaten”
- VI “Rotation kartesischer 2D-Koordinaten”

- VI “Verschiebung kartesischer 3D-Koordinaten”
- VI “Rotation kartesischer 3D-Koordinaten (Euler)”
- VI “Rotation kartesischer 3D-Koordinaten (Richtung)”
- VI “Umwandlung von 3D-Koordinaten”
- VI “Richtungscosinus in Euler-Winkel”
- VI “Euler-Winkel in Richtungscosinus”

Auf der Unterpalette **Hyperbelfunktionen** gibt es folgende neue Funktionen:

- Funktion “Cosecans Hyperbolicus”
- Funktion “Cotangens Hyperbolicus”
- Funktion “Secans Hyperbolicus”
- Funktion “Areacosecans”
- Funktion “Areacotangens”
- Funktion “Areasecans”

Auf der Unterpalette **Testen von Hypothesen** gibt es folgende neue VIs:

- VI “Korrelationstest”
- VI “Vorzeichentest”
- VI “T-Test”
- VI “Wilcoxon-Vorzeichen-Rangtest”
- VI “Z-Test”

Die Unterpalette **Integ & Diff** enthält das neue VI “Lobatto-Quadratur”.

Auf der Unterpalette **Interp & Extrap** gibt es folgende neue VIs:

- VI “Interpolationspolynom erzeugen”
- VI “Netzgitter (2D) erzeugen”
- VI “Interpolationspolynom entwickeln”
- VI “1D-Hermite-Interpolation”
- VI “1D interpolieren”
- VI “1D-Fourier interpolieren”
- VI “2D interpolieren”
- VI “Geordnete Tabelle durchsuchen”
- VI “1D-Spline-Interpolation”

Auf der Unterpalette **Numerisch** gibt es folgende neue Funktionen:

- Konstante “Maschinengenauigkeit”
- Funktion “Rechteck”

Auf der Unterpalette **Optimierung** gibt es folgende neue VIs:

- VI “Eingeschränkte nichtlineare Optimierung”
- VI “Quadratische Programmierung”
- VI “Optimierung ohne Nebenbedingungen”

Auf der Unterpalette **Polynom** gibt es das neue VI “Polynomkurve”.

Auf der Unterpalette **Wahrscheinlichkeit** gibt es folgende neue VIs:

- VI “Stetige Verteilung”
- VI “Stetige inverse Verteilung”
- VI “Stetige Momente”
- VI “Stetige PDF”
- VI “Stetige Zufallswerte”
- VI “Diskrete Verteilung”
- VI “Diskrete inverse Verteilung”
- VI “Diskrete Momente”
- VI “Diskrete PF”
- VI “Diskrete Zufallswerte”

Auf der Unterpalette **Wahrscheinlichkeit und Statistik** gibt es folgende neue VIs:

- VI “Korrelationskoeffizient”
- VI “Korrelationskoeffizient (Kendalls Tau)”
- VI “Korrelationskoeffizient (Spearman)”
- VI “Kovarianzmatrix”
- VI “Mittelwertmessungen”
- VI “Streuwertmessungen”
- VI “Perzentile”

Auf der Unterpalette **Trigonometrische Funktionen** gibt es folgende neue Funktionen:

- Funktion “Arcuscosecans”
- Funktion “Arcuscotangens”
- Funktion “Arcussecans”

Palette “Signalverarbeitung”

Die Unterpalette **Filter** enthält das neue VI “Nullphasenfilter”.

Die Unterpalette **Signaloperation** enthält das neue VI “Mischen”.

Auf der Unterpalette **Transformationen** gibt es folgende neue VIs:

- VI “Chirp-Z-Transformation”
- VI “DCT”
- VI “DST”
- VI “Inverse DCT”
- VI “Inverse DST”

Auf der Unterpalette **Signalverlaufskonditionierung** befindet sich das neue VI “Kontinuierliche Faltung (FIR)”.

Auf der Unterpalette **Signalverlaufsmessungen** ist das neue VI “Mehrere Frequenzen extrahieren” enthalten.

Auf der Unterpalette **Fenster** gibt es folgende neue VIs:

- VI “Blackman-Nuttall-Fenster”
- VI “Chebyshev-Fenster”
- VI “Gaußsches Fenster”
- VI “Symmetrisches Fenster”
- VI “Fenstereigenschaften”

Palette “Programmierung”

Auf der Unterpalette **Vergleich** gibt die neue Funktion “Leeres Array?”

Auf der Unterpalette **Komplex** gibt es folgende neue Funktionen:

- Funktion “Polar nach Re/Im”
- Funktion “Re/Im nach Polar”

Verbesserungen an den VIs für digitale Signalverläufe

Mit den VIs für digitale Signalverläufe können Abschnitte eines digitalen Signalverlaufs (DWDT) oder einer digitalen Tabelle (DTbl) ersetzt, digitale Signalverläufe oder Digitaldaten in analoge Signalverläufe umgewandelt und Binärdaten, analoge Signalverläufe, Digitaldaten oder Tabellen-Strings in digitale Signalverläufe oder Binärdaten umgewandelt werden. Mit Hilfe des VIs “Digitales Muster erzeugen” lässt sich ein digitales Muster mit verschiedenen Füllmustern erzeugen. Die Konstanten “Digital-

daten leer” und “Leerer digitaler Signalverlauf” geben leere Daten eines digitalen Signalverlaufs oder leere digitale Daten aus, was für die Initialisierung von Schieberegistern und das Erstellen von Signalverläufen wichtig ist.

Verbesserungen und Veränderungen an VIs

Folgende VIs für den digitalen Signalverlauf wurden in LabVIEW 8.0 verbessert bzw. verändert:

- Das VI “Digital: Invertieren” hat zwei neue Eingänge. Mit **Signalindex** wird das Signal festgelegt, an dem mit dem Invertieren der Daten begonnen wird, und unter **Anzahl der Signale** wird die Anzahl der zu invertierenden Signale angegeben.
- Am Eingang **Vergleichsmodus** des VIs “Nach Digitalmustern suchen” wird festgelegt, wie Werte von X für die Suche bearbeitet werden sollen.
- Das VI “Analog nach Digital” ist nun polymorph und kann einen analogen Signalverlauf in digitale Daten umwandeln.
- Das VI “Digital nach Analog” ist nun polymorph und kann digitale Daten in einen analogen Signalverlauf umwandeln.

Neue VIs und Funktionen

In LabVIEW 8.0 werden folgende neue VIs und Funktionen für digitale Signalverläufe eingeführt:

- VI “Binär nach Digital”
- VI “Digitales Muster erzeugen”
- Konstante “Digitaler Ring”
- VI “Digital nach Binär”
- Konstante “Digitaldaten leer”
- Konstante “Leerer digitaler Signalverlauf”
- VI “Abschnitt ersetzen”
- VI “Tabellen-String nach Digitaldaten”

Deaktivierungsstrukturen

Mit Hilfe der “bedingten Deaktivierungsstruktur” können Sie Blockdiagrammabschnitte deaktivieren, indem Sie jedem Unterdiagramm der Struktur einen Ausdruck zuweisen, der aus Symbolen und Werten zusammengesetzt ist (sowohl benutzerdefiniert als auch LabVIEW-definiert). Diese werden dann von LabVIEW ausgewertet, um zu bestimmen, welches Unterdiagramm verwendet werden soll.



Hinweis Nicht ausgeführte Blockdiagrammabschnitte werden auch nicht kompiliert. So zum Beispiel wird ein Blockdiagrammabschnitt, der sich im inaktiven Unterdiagramm der bedingten Deaktivierungsstruktur befindet, nicht kompiliert. Ebenso wenig wird der Blockdiagrammabschnitt in einer Case-Struktur kompiliert, der mit einer Konstante verbunden ist.

Mit Hilfe der “Diagrammdeaktivierungsstruktur” lassen sich Teile des Blockdiagramms bestimmen, die nicht kompiliert werden sollen.

Verbesserungen an den Express-VIs

Express-VIs sind für allgemeine Arten von Messungen geeignet.

Verbesserungen und Veränderungen an VIs

Folgende Express-VIs wurden in LabVIEW 8.0 verbessert bzw. verändert:

- Das Express-VI “Vergleich” hat zwei neue Eingänge. Mit Hilfe von **Zu vergleichende Objekte** lassen sich Datenpunkte, Zeitstempel, die Zeit zwischen Punkten, Anzahl der Datenpunkte und Signalnamen des Eingangssignals miteinander vergleichen. **Konstanter Vergleichswert** legt den konstanten Wert fest, mit dem der Eingang **Operand 1** verglichen werden soll. Dieses VI wurde optimiert. Es gibt nun mehr geeignete Ergebnisse im Falle eines leeren ersten Eingangssignals aus.
- Die Express-VIs “LabVIEW-Messdaten aus Datei lesen” und “LabVIEW-Messdaten in Datei schreiben” wurden in “Messwerte aus Datei lesen” und “Messwerte in Datei schreiben” umbenannt. Diese Express-VIs arbeiten sowohl mit Binärdateien für Messwerte (* . tdm) als auch mit Textdateien für Messwerte (* . lvm). Mit Hilfe der Option **Binär (TDM)** im Konfigurationsdialogfeld wird das Format auf Binärdatei eingestellt.
- Die Express-VIs “Messdaten aus Datei lesen”, “Messwerte in Datei schreiben”, “Eingabeaufforderung für Benutzer” und “Bericht” richten sich in ihrer Funktion nach dem Zielsystem. Wenn (möglicherweise) kein Host-Computer mit dem aktuellen Zielsystem verbunden ist, werden in den Konfigurationsdialogfeldern neben Optionen, die ohne Host ungültig sind, Warnungen angezeigt. Diese VIs geben auch Fehlermeldungen aus, wenn sie auf eine Eingabeaufforderung konfiguriert sind und auf einem System ohne Benutzeroberfläche ausgeführt werden (zum Beispiel LabVIEW Real-Time Module ohne angeschlossenen Host-Computer).

Wird das VI “Bericht” auf einer anderen Plattform als Windows ausgeführt, kann im Konfigurationsdialogfeld als Ausgabeart nur **HTML für Web-Seite** ausgewählt werden.

- Das Express-VI “Relais” arbeitet mit der Einstellung am Eingang **Aktivieren**, wenn kein Wert mit dem **Aktivieren**-Eingang im Blockdiagramm verbunden ist.
- Am **Auswahleingang** des Express-VIs “Signale auswählen” kann ein Signalabschnitt vom Eingang **Signale** ausgewählt werden.
- Die Express-VIs “Daten schreiben” und “Messwerte in Datei schreiben” wurden dahingehend geändert, dass mehrmals vergebene Kanalnamen automatisch einen individuellen Namen erhalten. Wenn sich ein Datei- oder ein Kanalname wiederholt, wird an das Ende des Namens eine Zahl angehängt. Wenn Sie zum Beispiel die Namen Sinus, Sinus, Rechteck, Rechteck und Sinus vergeben, werden diese Namen auf Sinus, Sinus 1, Rechteck, Rechteck 1 und Sinus 2 geändert.

Neue VIs und Funktionen

In LabVIEW 8.0 werden folgende neue Express-VIs eingeführt:

- Express-VI “Audioaufnahme”
- Express-VI “Zweikanal-Spektrummessung”
- Express-VI “Audioausgabe”
- Express-VI “DIAdem-Report”

Express-VI “DIAdem-Report”

Mit Hilfe des Express-VIs “DIAdem-Report” können Sie direkt aus LabVIEW-Daten einen Report erstellen. Sie können diesen Report als PDF- oder HTML-Datei exportieren, ausdrucken oder automatisch anzeigen lassen. Der Report wird in LabVIEW vollständig konfiguriert.



Hinweis Für das Express-VI “DIAdem-Report” muss mindestens DIAdem 9.1 Service Pack 2 installiert sein. Besuchen Sie zum Herunterladen einer Evaluierungsversion bzw. zum Kauf von DIAdem die National-Instruments-Website unter ni.com.

Verbesserungen an den Datei-I/O-VIs und -Funktionen

Auf der Palette Datei-I/O befinden sich VIs und Funktionen, die dahingehend verbessert wurden, dass sie das Lesen von Daten bzw. das Schreiben von Daten in Dateien einfacher machen. Darüber hinaus enthalten diese VIs und Funktionen einen verbesserten Pfadeingang und neue Datenprotokollfunktionen. Auf der Palette **Datei-I/O** sind auch neue VIs enthalten, die Sie zur Arbeit mit Zip-Dateien verwenden können.

Pfadeingang

Datei-I/O-Funktionen, die mit Pfaden oder Refnums arbeiten, haben einen Eingang, an dem (`Dialog`, wenn leer) steht, zum Beispiel den Eingang **Dateipfad (Dialog, wenn leer)**. Ist dieser Eingang nicht verbunden, wird ein Dialogfeld angezeigt, in dem der Benutzer aufgefordert wird, während der Ausführung der Funktion eine Datei oder ein Verzeichnis auszuwählen. Jede dieser Funktionen hat einen Ausgang mit dem Namen **abgebrochen**. Wenn der Benutzer im Dialogfeld auf die Schaltfläche **Abbrechen** klickt, wird unter **abgebrochen** TRUE ausgegeben und die Funktion gibt eine Fehlermeldung aus. Wenn Sie einen Pfad mit einer Datei-I/O-Funktion verbinden, öffnet die Funktion die Datei mit den Mindestzugriffsrechten, die für eine bestimmte Operation nötig sind, und die Datei muss nicht mehr manuell geöffnet oder geschlossen werden.

Datenprotokoll-Funktionen

Mit Hilfe der Datenprotokoll-Funktionen lassen sich Daten in Datenprotokolldateien schreiben oder aus diesen auslesen. Auf der Palette Datenprotokoll gibt es folgende neue Funktionen:

- Funktion “Öffnen/Erstellen/Ersetzen einer Protokolldatei”
- Funktion “Position im Datenprotokoll ermitteln”
- Funktion “Anzahl der Datensätze ermitteln”
- Funktion “Datenprotokoll lesen”
- Funktion “Position im Datenprotokoll festlegen”
- Funktion “Anzahl der Datensätze festlegen”
- Funktion “Datenprotokoll schreiben”

Datei-I/O-VIs und -Funktionen

Die Datei-I/O-VIs und -Funktionen dienen zum Öffnen und Schließen von Dateien, Datenaustausch mit Dateien, Erstellen von Verzeichnissen und Dateien, Abrufen von Angaben zu einem Verzeichnis und Übertragen von Strings, Zahlen, Arrays und Clustern in Dateien. Auf der Palette Datei-I/O gibt es folgende neue Funktionen:

- Funktion “Öffnen/Erstellen/Ersetzen einer Datei”
- Funktion “Aus Binärdatei lesen”
- Funktion “Aus Textdatei lesen”
- Funktion “In Binärdatei schreiben”
- Funktion “In Textdatei schreiben”

Zip-VIs

Mit Hilfe der Zip-VIs lassen sich neue Zip-Dateien erstellen, Dateien zu Zip-Dateien hinzufügen und Zip-Dateien schließen.

Verschiedene Neuerungen und Änderungen

In LabVIEW 8.0 gibt es folgende Verbesserungen und Veränderungen an den Datei-I/O-VIs und -Funktionen:

- Die Funktion “Kopieren” hat nun Eingänge mit dem Namen **überschreiben** und **Eingabeaufforderung** sowie einen Ausgang mit dem Namen **abgebrochen**.
- Die Funktion “Löschen” hat einen neuen Eingang mit dem Namen **ganze Hierarchie**. Ist dieser Eingang TRUE wird die gesamte Hierarchie gelöscht, die in oder unter **Pfad** festgelegt wurde.
- Die Funktion “Dateidialog” ist nun das Express-VI “Dateidialog”. Zum Einstellen der Dateien oder Verzeichnisse, die die Benutzer im Dateidialogfeld auswählen können, klicken Sie doppelt auf dieses VI, so dass sich das Dialogfeld **Auswahlmodus** öffnet.
- Die Funktionen “Datei schließen”, “Kopieren”, “Ordner erstellen”, “Löschen”, “Datei-/Verzeichnisinformation”, “Datei entleeren” und “Verschieben” haben neue Anschlussfeldkonfigurationen. Diese neuen Konfigurationen können die Lage von Verbindungen auf dem Blockdiagramm beeinflussen, die in früheren LabVIEW-Versionen erstellt wurden.
- Mit Hilfe der Funktionen “Verschieben” und “Kopieren” lassen sich vorhandene Dateien und Verzeichnisse überschreiben.
- Die Funktion “Datei-/Verzeichnisinformation” hat zwei neue Eingänge. Unter **Verknüpfung** wird angegeben, ob die Datei oder das Verzeichnis eine Verknüpfung ist. Am Eingang **Erkannter Pfad** wird der Pfad zur Datei oder zum Verzeichnis ausgegeben, mit der bzw. dem die Verknüpfung besteht.
- Die String-Instanz des VIs “Schlüssel lesen” hat nicht mehr den Parameter **Multibyte-Codierung (0: Keine)**.

Verbesserungen am Assistenten zur Instrumenten-I/O

Beim Instrumenten-I/O-Assistenten gibt es folgende Verbesserungen:

- Sie können das Zeichen, mit dem das letzte Zeichen eines Befehls für ein Gerät bestimmt wird, anhand des Pulldown-Menüs **Endzeichen** im Schritt **Gerät auswählen** festlegen. Sie können auch **<benutzerdefiniert>** zur Eingabe eines Endzeichens in das Textfeld **Endzeichen** auswählen.

- Zum Hinzufügen eines Eingangsparameters zu einem Gerätebefehl wählen Sie aus dem Textfeld **Befehl eingeben** im Schritt **Schreiben** den gewünschten Befehl aus und betätigen Sie die Schaltfläche **Parameter hinzufügen**. Stattdessen können Sie aber auch den Befehl mit der rechten Maustaste anklicken und aus dem Kontextmenü **Parameter hinzufügen** auswählen.
- Sie können den Schritt **Abfragen und analysieren** auch in die beiden Schritte **Schreiben** und **Lesen und Analyse** aufspalten, um die parametrisierten Eingänge verwenden zu können. Klicken Sie zum Aufteilen des Schritts **Abfragen und Analysieren** mit der rechten Maustaste auf den Schritt im Fenster **Schritt-Sequenz** und wählen Sie **Schritt aufteilen in Schritte Schreiben und Lesen** aus dem Kontextmenü.
- Sie können den Assistenten zur Instrumenten-I/O so konfigurieren, dass alle Daten in einem Token analysiert werden, bis keine Daten mehr vorhanden sind. Um den Assistenten zur Instrumenten-I/O so zu konfigurieren, dass alle Daten analysiert werden, wählen Sie beim Schritt **Abfragen und Analysieren** aus dem Pulldown-Menü **Anzahl** die Option **Nach Ende der Daten** aus oder klicken Sie auf die Schaltfläche **Nach Ende der Daten** unter dem Pulldown-Menü **Anzahl**. Die Daten werden auch bis zum Ende analysiert, wenn Sie auf die Schaltfläche **automatische Analyse** klicken. Die Option **Bis Datenende** ist nur dann verfügbar, wenn es sich beim eingestellten Token um das letzte Token handelt.

Verbesserungen an den Audio-VIs

Die Palette **Audio** enthält VIs zur Audioausgabe und -aufnahme.



Hinweis (Windows) Für die Audio-VIs muss eine DirectX-Version ab 8.0 installiert sein.

Diese VIs wurden auf folgende Weise verbessert:

- Die VIs arbeiten mit Audiodaten bis zu 24 Bit.
- Die VIs arbeiten mit Audiodateien mit mehreren Kanälen und sind für Mono und Stereo geeignet.
- Audiodaten werden durch einen Signalverlauf dargestellt. Die Daten des Y-Arrays können in Form von vorzeichenlosen 8-Bit-, vorzeichenbehafteten 16-Bit- oder vorzeichenbehafteten 32-Bit-Integern sowie mit Datentypen einfacher und doppelter Genauigkeit dargestellt werden. Jeder Signalverlauf steht für einen Kanal.
- Die Audiodaten werden durch Puls-Code-Modulation digitalisiert.
- Die VIs sind zur fortlaufenden Audioausgabe in der Lage.
- Mit den VIs ist Streaming der Wave-Dateien möglich.
- Die VIs sind in Bezug auf die Fehlerprüfung verbessert worden.

Verbesserungen an den String-Funktionen

Anhand der Funktionen “Regulären Ausdruck suchen” und “Muster suchen und ersetzen” können mit der PCRE-Syntax (Perl Compatible Regular Expression) komplexere reguläre Ausdrücke gesucht werden. Um die Funktion für die fortgeschrittene Suche nach regulären Ausdrücken zu konfigurieren und den Suchtreffer ersetzen zu lassen, klicken Sie mit der rechten Maustaste auf die Funktion “String suchen und ersetzen” und wählen Sie aus dem Kontextmenü die Option **Regulärer Ausdruck** aus. Zur Anzeige aller Treffer, die die Funktion “Regulären Ausdruck suchen” erzielt hat, müssen Sie die Funktion möglicherweise aufziehen. Eine Suche nach regulären Ausdrücken ist auch mit der Funktion “Muster suchen” oder “Regulären Ausdruck suchen” möglich. Die Funktion “Regulären Ausdruck suchen” hat mehr Optionen, ist aber bedeutend langsamer als die Funktion “Muster suchen”.

Neuerungen beim VI-Hierarchie-Fenster

Wählen Sie **Anzeigen»VI-Hierarchie**, um das Fenster **VI-Hierarchie** zu öffnen. Darin werden geöffnete LabVIEW-Projekte und verwendete Systeme sowie die Aufrufhierarchie für alle VIs im Speicher angezeigt. Das Symbol am oberen Ende der **VI-Hierarchie** steht für die Hauptapplikation in LabVIEW; darunter befinden sich alle geöffneten VIs, die sich nicht in einem Projekt befinden. Im Fenster **VI-Hierarchie** wird jedes Projekt im Speicher durch ein bestimmtes Symbol gekennzeichnet. Alle weiteren Zielsysteme werden unter dem Projekt angezeigt.

Verbesserungen am Frontpanel

In LabVIEW 8.0 gibt es die nachfolgenden Verbesserungen und Änderungen am Frontpanel.

Änderungen am Graphen-Cursor

Im Hinblick auf Graphen-Cursor gibt es bei LabVIEW 8.0 folgende Änderungen:

- Klicken Sie zum Erstellen eines Cursors mit der rechten Maustaste auf eine beliebige Stelle der Cursor-Legende, wählen Sie **Cursor erstellen** und anschließend einen Cursor-Modus aus dem Kontextmenü aus. Es gibt folgende Cursor-Modi:

Tabelle 4. Änderungen an den Cursor-Modi in LabVIEW 8.0

Cursor-Modus in LabVIEW 7.1	Cursor-Modus in LabVIEW 8.0	Beschreibung des Cursor-Modus in LabVIEW 8.0
Frei	Frei	Bewegt den Cursor unabhängig von den Plots frei über den Plot-Bereich.

Tabelle 4. Änderungen an den Cursor-Modi in LabVIEW 8.0 (Fortsetzung)

Cursor-Modus in LabVIEW 7.1	Cursor-Modus in LabVIEW 8.0	Beschreibung des Cursor-Modus in LabVIEW 8.0
An Punkt einrasten und An Plot einrasten	Einzel-Plot	Rastet den Cursor nur an dem Plot ein, zu dem der Cursor gehört. Der Cursor kann nur entlang des zugehörigen Plots bewegt werden. Um Plots mit einem Cursor zu verknüpfen, klicken Sie mit der rechten Maustaste auf eine Zeile der Cursor-Legende und wählen Sie eine Option für Einrasten auf aus.
—	Multi-Plot	Rastet den Cursor nur an einem bestimmten Punkt im Plotbereich ein. Beim Multi-Plot-Cursor werden an einem bestimmten x-Wert die Werte für alle Plots angezeigt, für die der Cursor gilt. Sie können den Cursor an jedem beliebigen Plot im Plotbereich einrasten. Um Plots mit einem Cursor zu verknüpfen, klicken Sie mit der rechten Maustaste auf eine Zeile der Cursor-Legende und wählen Sie eine Option für Einrasten auf aus. Dieser Modus ist nur für Graphen mit verschiedenen Signaltypen gültig.

- Nach dem Erstellen kann der Modus eines Cursors nicht mehr verändert werden. Sie müssen sonst den Cursor löschen und einen neuen erstellen. Klicken Sie zum Löschen eines Cursors mit der rechten Maustaste auf eine Zeile der Cursor-Legende und wählen Sie aus dem Kontextmenü die Option **Cursor löschen** aus.
- Die Schaltflächen **Cursor-Bewegung**, **Formatierung** und **Fixierung** werden nicht auf der rechten Seite der Cursor-Legende angezeigt. Klicken Sie mit der rechten Maustaste auf eine Zeile der Cursor-Legende und wählen Sie die gewünschten Optionen zum Anpassen des Cursors aus dem Kontextmenü aus. Alle Optionen, zu denen Sie in früheren LabVIEW-Versionen über die Schaltflächen in der Cursor-Legende gelangt sind, befinden sich nun im Kontextmenü.
- Mit Hilfe der Eigenschaft “x-Achse” können Sie die x-Achse des Cursors programmatisch festlegen. Diese Eigenschaft gibt es nur, wenn der Cursor-Modus **Frei** lautet. Diese Eigenschaft ist vergleichbar mit dem Menüpunkt **x-Achse** im Kontextmenü der Cursor-Legende eines Graphen.

- Mit Hilfe der Eigenschaft “y-Achse” können Sie die y-Achse des Cursors programmatisch festlegen. Diese Eigenschaft gibt es nur, wenn der Cursor-Modus **Frei** lautet. Diese Eigenschaft ist vergleichbar mit dem Menüpunkt **y-Achse** im Kontextmenü der Cursor-Legende eines Graphen.
- Um die Werte von Plots programmatisch zu verfolgen, werden die Eigenschaften “Plots betrachten” und “Alle Plots betrachten” verwendet. Diese Eigenschaften gelten nur für Cursor, die zu mehreren Plots gehören.

Notizen für Graphen

Mit Hilfe von Notizen an einem Graphen können Sie bestimmte Werte von Plots hervorheben. Eine Notiz hat immer eine Beschriftung und einen Pfeil auf den Wert, für den die Notiz gilt. Sie können das Aussehen der Notizen ändern und festlegen, wie die Notiz am Plot einrasten soll. Zum Erstellen einer Notiz klicken Sie den Graphen mit der rechten Maustaste an und wählen Sie **Datenoperationen»Notiz erstellen** aus dem Kontextmenü. Daraufhin wird das Dialogfeld **Notiz erstellen** angezeigt. Um Einstellungen an einer Notiz vorzunehmen, klicken Sie mit der rechten Maustaste auf die Notiz und wählen Sie eine der Optionen aus. Anhand der Eigenschaft “Notizenliste” können Sie sich programmatisch ein Array mit Informationen zu allen Notizen im Plotbereich ausgeben lassen.

Anzeige von Grafiken im Plotbereich von Graphen

Grafiken können im Hintergrund, Vordergrund oder in der Mitte des Plotbereichs von Graphen dargestellt werden. Verwenden Sie dazu die Eigenschaften “Plot-Bilder:Hinten”, “Plot-Bilder:Vorn” und “Plot-Bilder:Mitte” des Signalverlaufsgraphen, Intensitätsgraphen oder Mischsignal-Graphen. Die Eigenschaft “Plot-Bilder:Vorn” gilt nur für Intensitätsgraphen. Die Daten des eigentlichen Plots befinden sich zwischen dem Vordergrund und dem mittleren Plot-Bild. Die Gitterlinien befinden sich zwischen dem mittleren Plot-Bild und dem Hintergrund. Um ein vorhandenes Bild aus dem Hintergrund, Vordergrund oder dem mittleren Plotbereich zu entfernen, müssen Sie ein leeres Bild mit der entsprechenden Eigenschaft verbinden oder das Bild transparent machen.



Hinweis Wenn Sie die Größe eines Graphen ändern, passen sich die Bilder des Plotbereichs nicht automatisch an. Das VI muss neu gestartet werden, so dass es die Bilder noch einmal anzeigt.

Zur ordnungsgemäßen Anzeige der Bilder müssen die Koordinaten des Graphen in Frontpanel-Koordinaten umgerechnet werden. Mit Hilfe der Methode “XY in Koordinaten umrechnen” lassen sich die Koordinaten des Graphen oder des Diagramms in Frontpanel-Koordinaten umrechnen. Es

werden jedoch nur der linke und der obere Rand des Plots angepasst, so dass der Plot um den Koordinatenursprung des Frontpanels herum angeordnet wird.

Beispiele zu den Eigenschaften der Gruppe “Plot-Bilder” und zur Methode “XY in Koordinaten umrechnen” finden Sie im Verzeichnis `labview\examples\general\graphs\Graph Pictures.llb`.

Exportieren von Graphen-, Diagramm- und Tabellenabbildungen

Eine Abbildung von Graphen, Diagrammen, Tabellen, Binärtabellen sowie Bedien- und Anzeigeelementen für digitale Signalverläufe lässt sich in einem der folgenden Formate exportieren:

(Windows) *.emf, *.bmp oder *.eps

(Mac OS) *.pict, *.bmp oder *.eps

(Linux) *.bmp oder *.eps

Zum Exportieren einer Abbildung klicken Sie mit der rechten Maustaste auf ein Bedienelement und wählen Sie aus dem Kontextmenü die Option **Datenoperationen»Vereinfachtes Bild exportieren** aus. Wählen Sie im Dialogfeld **Vereinfachtes Bild exportieren** das Format aus und geben Sie an, ob das Bild in der Zwischenablage oder auf einem Datenträger gespeichert werden soll. Das programmatische Exportieren von Abbildungen ist mit der Methode “Bild exportieren” möglich.

Anzeigen von Ebenen in XY-Graphen

Im XY-Graphen können Nyquist-, Nichols-, S- und Z-Ebenen angezeigt werden. Zum Anzeigen dieser Ebenen klicken Sie mit der rechten Maustaste auf den XY-Graphen und wählen Sie **Optionale Ebene**. Der Menüpunkt **Optionale Ebene** enthält die verschiedenen Arten von Ebenen, die in einem XY-Graphen angezeigt werden können, wie die Nyquist-, Nichols-, S- oder Z-Ebene. Wenn Sie eine optionale Ebene auswählen, enthält das Menü **Optionale Ebene** Menüpunkte zum Ein- und Ausblenden der Linien und Beschriftungen der Ebenen sowie der kartesischen Linien.

Sie können die Eigenschaft “Optionale Ebene:Ebenentyp” auch zum programmatischen Ein- bzw. Ausblenden dieser Ebenen im XY-Graphen verwenden. Die Linien und Beschriftungen einer Ebene werden programmatisch mit den Eigenschaften “Optionale Ebene:Linien sichtbar” und “Optionale Ebene:Beschriftungen sichtbar” ein- und ausgeblendet. Zum programmatischen Ein- und Ausblenden der kartesischen Linien in einem XY-Graphen wird die Eigenschaft “Kartesische Linien:Sichtbar” verwendet.

XY-Graphen arbeiten auch mit Arrays aus komplexen Werten. In diesem Fall wird der Realteil auf der x-Achse und der Imaginärteil auf der y-Achse dargestellt.

Mischsignal-Graph

In Mischsignal-Graphen lassen sich Daten unterschiedlichen Typs darstellen, die eine gemeinsame x-Achse haben. Mischsignal-Graphen arbeiten mit den gleichen Datentypen wie Signalverlaufsgraphen, XY-Graphen und digitale Signalverlaufsgraphen. Darüber hinaus können alle Cluster als Eingang anliegen, die eine Kombination der genannten Datentypen enthalten.

Wenn Sie mehrere Plotbereiche zu einem Mischsignal-Graphen hinzufügen, wird jeder Plotbereich mit seiner eigenen y-Achse angezeigt. Der Mischsignal-Graph erstellt bei Bedarf automatisch Plotbereiche zur Anzeige von Kombinationen aus analogen und digitalen Daten.

Die Plotlegende am Mischsignal-Graphen besteht aus Baumstruktur-Elementen und wird links neben dem Plotbereich des Graphen angezeigt. Jedes Baumstruktur-Element stellt einen Plotbereich dar. Plots werden zwischen den Plotbereichen mit Hilfe der Plotlegende verschoben.

Durch Erstellen eines Multi-Plot-Cursors an einem Mischsignal-Graphen können Werte von mehreren Plots an einem bestimmten x-Wert kenntlich gemacht werden.

Tastatursteuerung

Zur Anzeige der Seite **Tastatursteuerung** im Dialogfeld **Eigenschaften** klicken Sie mit der rechten Maustaste auf ein Bedienelement auf dem Frontpanel und wählen Sie aus dem Kontextmenü die Option **Fortgeschritten»Tastatursteuerung** aus. Sie können diese Seite auch öffnen, indem Sie ein Bedienelement mit der rechten Maustaste anklicken, aus dem Kontextmenü die Option **Eigenschaften** auswählen und die Registerkarte **Tastatursteuerung** anklicken. Auf der Seite **Tastatursteuerung** wurden folgende Änderungen vorgenommen:

- Der Abschnitt **Tastenbelegung** wurde in **Fokus** umbenannt. Unter **Fokus** sind nun die Tasten <Entf>, <Einf>, <Mute>, <Play>, <Volume Up>, <Volume Down> sowie die Funktionstasten F13 bis F24 aufgeführt.
- Der Abschnitt **Aktuelle Zuordnungen** wurde in **Vorhandene Zuordnungen** umbenannt.
- Auf der Seite **Tastatursteuerung** gibt es einen Abschnitt **Umschaltmodus**, mit dem Tastenkombinationen zum Hin- und Herschalten eines booleschen Bedienelementen zugewiesen werden können. Eine

Tastenkombination zum Umschalten eines booleschen Bedienelement kann auch mit der Eigenschaft “Umschalttaste” festgelegt werden.

- Auf der Seite **Tastatursteuerung** gibt es die Abschnitte **Inkrementieren** und **Dekrementieren**, in denen Tastenkombinationen zum Erhöhen und Verringern der Werte in numerischen Elementen, Ring- und Enum-Elementen sowie Schiebe- und Drehreglern eingestellt werden können. Sie können jedoch auch die Eigenschaften “Aufwärtstaste” und “Abwärtstaste” verwenden, um Tasten zum Erhöhen oder Verringern eines Werts anzugeben.

Die Eigenschaft “Tastatursteuerung” wurde in “Zugewiesene Taste” umbenannt. Bei booleschen Bedienelementen ist dies jedoch nur die Taste, mit der der Fokus auf das Element gelegt wird. Zum Festlegen der Taste, mit der das Element umgeschaltet wird, benötigen Sie die Eigenschaft “Umschalttaste”.

Mindestgröße des Frontpanels

In LabVIEW 7.x und Vorgängerversionen bezieht sich die Mindestgröße des Frontpanels auf den Inhaltsbereich ohne die Bildlaufleisten. In LabVIEW 8.0 bezieht sich die Mindestgröße bei Frontpaneln mit nur einem Fensterbereich auf den Inhalt des Bereichs. Die Bildlaufleisten sind nicht eingeschlossen. Bei Frontpaneln mit mehreren Fensterbereichen bezieht sich die Mindestgröße dagegen auf das ganze Frontpanel, alle sichtbaren Bildlaufleisten eingeschlossen.

Trennbalken und Fensterbereiche

Anhand der Trennbalken aus der Palette **Container** können Sie das Frontpanel in zwei oder mehrere Bereiche unterteilen, durch die Sie separat scrollen können. Mit diesen Fensterbereichen können Sie eine Benutzeroberfläche mit verschiedenen Abschnitten gestalten. Sie können durch Trennbalken aber auch einen Fensterbereich als Symbolleiste oder Statusleiste kennzeichnen.

Zum programmatischen Einteilen des Fensters in Bereiche werden die Eigenschaften, Methoden und Ereignisse für Fensterbereiche verwendet.

Trennbalken werden programmatisch mit den Eigenschaften und Ereignissen für Trennbalken eingefügt.

In LabVIEW 8.0 besteht die Option **Bildlaufleisten anzeigen** aus dem Dialogfeld **Fenstererscheinungsbild anpassen**, die es in LabVIEW 7.x und Vorgängerversionen gab, aus den Optionen **Horizontale Bildlaufleiste anzeigen** und **Vertikale Bildlaufleiste anzeigen**. Wenn Sie einen Trennbalken zum Frontpanel hinzufügen, können diese Optionen nicht geändert

werden. Die Optionen spiegeln die Einstellungen im Feld ganz oben links wider, wenn sich Trennbalken auf dem Frontpanel befinden.

Individuelle Gestaltung von Laufzeit-Kontextmenüs

Das Laufzeit-Kontextmenü kann für jedes Bedienelement, das in einem VI enthalten ist, individuell gestaltet werden. Klicken Sie zur Anzeige des Dialogfelds **Kontextmenü-Editor** mit der rechten Maustaste auf das Element und wählen Sie **Fortgeschritten»Laufzeit-Kontextmenü»Bearbeiten**. Anschließend können Sie das Kontextmenü bearbeiten. In diesem Dialogfeld können Sie einem Element das Standard-Kontextmenü oder eine eigene Kontextmenü-Datei (*.rtm) zuweisen. Durch das Ereignis “Kontextmenü-Aktivierung?” sowie die Funktionen “Menüpunkte einfügen” und “Menüeinträge löschen” lassen sich Kontextmenüs programmatisch individuell anpassen.

Sie können einem Frontpanel auch Kontextmenüs hinzufügen. Dazu werden die Ereignisfelder **Kontextmenü-Aktivierung?** und **Kontextmenü-Auswahl** verwendet.

Sie können auch mit der rechten Maustaste auf ein Bedienelement klicken und aus dem Kontextmenü die Option **Fortgeschritten»Laufzeit-Kontextmenü»Deaktivieren** auswählen, um das Laufzeitkontextmenü eines Elements zu deaktivieren.

Sie können die Standard-Kontextmenüs für alle Bedienelemente deaktivieren, indem Sie **Datei»VI-Einstellungen** und dann **Fenstererscheinungsbild** aus dem Pulldown-Menü **Kategorie** auswählen und anschließend auf die Schaltfläche **Anpassen** klicken und die Option **Standardkontextmenüs zur Laufzeit zulassen** deaktivieren. Selbst erstellte Kontextmenüs können jedoch immer eingeblendet werden.

Verwendung von XControls

Mit Hilfe der XControls können Sie komplexe Bedien- und Anzeigeelemente in LabVIEW entwerfen und erstellen. XControls enthalten die folgenden Funktionen:

- Sie können mit Hilfe der in LabVIEW enthaltenen Bedien- und Anzeigeelemente ein XControl erstellen und somit die Vorteile der kombinierten Funktion der LabVIEW-Anzeige und -Bedienelemente nutzen.
- Wenn Sie ein XControl in einem VI verwenden, wird das Blockdiagramm des entsprechenden VIs vereinfacht, da sich das XControl wie ein Bedienelement verhält.

- Mit Hilfe einer XControl-Bibliothek lassen sich Verhalten, Aussehen, Merkmale, Eigenschaften und Methoden eines XControls festlegen.
- Sie können das Kontextmenü eines XControls so anpassen, dass es sowohl neue Menüpunkte als auch die bereits vorhandenen Menüpunkte der Bedien- und Anzeigeelemente enthält, mit denen Sie das XControl erstellt haben.

Zum Erstellen eines XControls klicken Sie im Projekt-Explorer mit der rechten Maustaste auf **Mein Computer** und wählen Sie **Neu»XControl** aus dem Kontextmenü aus.



Hinweis XControls können nur im LabVIEW Professional Development System erstellt und bearbeitet werden. VIs mit XControls sind in allen LabVIEW-Paketen lauffähig.

Ein Anwendungsbeispiel für XControls finden Sie im Projekt “Simple Dual Mode Thermometer XControl” im Verzeichnis `labview\examples\general\xcontrols\Dual Mode Thermometer`.

Verwenden von .NET-Elementen

.NET-Steuerelemente auf dem Frontpanel können genauso verwendet werden wie ActiveX-Steuerelemente. Erstellen Sie ein neues Bedienelement in einem .NET-Container oder fügen Sie .NET-Steuerelemente der Elementpalette zum späteren Gebrauch hinzu. Verbinden Sie das .NET-Element im Blockdiagramm mit einem Eigenschaftsknoten, Methodenknoten oder dem Knoten “Ereignis-Callback registrieren”, um die Eigenschaften festzulegen, die Methoden aufzurufen oder die Ereignisse für das Bedienelement zu bearbeiten.

Fügen Sie zum Erstellen eines neuen .NET-Elements einen .NET-Container auf dem Frontpanel ein. Klicken Sie mit der rechten Maustaste auf den .NET-Container und wählen Sie **.NET-Element einfügen** aus dem Kontextmenü, um das Dialogfeld **.NET-Element auswählen** anzuzeigen. Wählen Sie das gewünschte Element aus und klicken Sie auf **OK**.

Zum Hinzufügen eines .NET-Elements zur **Elemente**-Palette wählen Sie **Werkzeuge»NET & ActiveX»NET-Elemente zur Palette hinzufügen**, um zum Dialogfeld **NET-Elemente der Palette hinzufügen** zu gelangen. Wählen Sie das gewünschte .NET-Element aus und klicken Sie auf **OK**. LabVIEW speichert die Elemente per Voreinstellung im Verzeichnis `labview\menus\Controls\DotNet & ActiveX`, da alle Dateien in diesem Verzeichnis automatisch auf der Palette **.NET & ActiveX** angezeigt werden.

Listenfelder, Baumstruktur-Elemente und Tabellen

In LabVIEW 8.0 wird ein erweiterter Funktionsumfang eingeführt. Dazu gehören neue Eigenschaften und Methoden für Listenfelder, Tabellen, digitale Bedienelemente und Baumstruktur-Elemente. Diese vier Arten von Bedienelementen haben viele gemeinsame Funktionen und Eigenschaften.

- Zur Anzeige der Zeilentitel klicken Sie mit der rechten Maustaste auf ein mehrspaltiges Listenfeld oder ein Baumstruktur-Element und wählen Sie aus dem Kontextmenü die Option **Sichtbare Objekte» Zeilentitel** aus. Zur programmatischen Anzeige der Zeilentitel können Sie auch die Eigenschaft “Sichtbare Objekte:Zeilentitel sichtbar” eines “mehrspaltigen Listenfelds” oder “Baumstruktur-Elements” verwenden.
- Zum Anzeigen der vertikalen und horizontalen Linien klicken Sie mit der rechten Maustaste auf ein mehrspaltiges Listenfeld, eine Tabelle oder ein Baumstruktur-Element und wählen Sie aus dem Kontextmenü entweder **Sichtbare Objekte»Vertikale Linien** oder **Sichtbare Objekte»Horizontale Linien** aus. Zur programmatischen Anzeige der vertikalen Linien für ein “mehrspaltiges Listenfeld”, eine “Tabelle” oder ein “Baumstruktur-Element” können Sie auch die Eigenschaft “Sichtbare Objekte:Vertikale Linien sichtbar” verwenden. Um die horizontalen Linien für ein “einspaltiges Listenfeld”, “mehrspaltiges Listenfeld”, eine “Tabelle” oder ein “Baumstruktur-Element” einzublenden, gibt es die Eigenschaft “Sichtbare Objekte:Horizontale Linien sichtbar”.
- Mit Hilfe des Farbwerkzeugs können Sie alle Zellen, Zeilentitel und Spaltentitel sowie die Spaltenüberschrift der linken oberen Zelle von Tabellen, Listenfeldern und Baumstruktur-Elementen unabhängig voneinander farbig gestalten.
- Passt ein Text nicht vollständig in eine Zelle, dann erscheint ein Hinweisstreifen mit dem kompletten Text, wenn das VI ausgeführt wird und Sie den Cursor des Bedienwerkzeugs über die Zelle bewegen.
- In Listenfeldern und Baumstruktur-Elementen gibt es vier neue Elementsymbole. Das Aussehen dieser Kontrollkästchen-Symbole richtet sich nach Ihrem Betriebssystem. Klicken Sie zum Anzeigen aller Symbole mit der rechten Maustaste auf ein Objekt im Bedienelement und wählen Sie im Kontextmenü die Option **Objektsymbol** aus. Listenfelder haben die gleichen Eigenschaften für benutzerdefinierte Symbole wie das Baumstruktur-Element.
- Drücken Sie die Tastenkombination <Shift-Enter>, damit in Listenfeldern oder Baumstruktur-Elementen unter dem aktuellen Element ein neues eingefügt wird. Bei Baumstrukturen wird das neue Element durch Drücken dieser Tastenkombination auf der Hierarchie-Ebene des aktuellen Elements eingefügt. Alle nachfolgenden Elemente werden um eine Zeile nach unten verschoben.

- Klicken Sie mit der rechten Maustaste auf ein Listenfeld bzw. die Baumstruktur und wählen Sie **Editierbare Zellen** aus dem Kontextmenü, so dass Zellen während der Laufzeit bearbeitet werden können. Daneben gibt es auch die Eigenschaft “Zellenbearbeitung zulassen” zur programmatischen Bearbeitung von Zellen in einspaltigen Listenfeldern, mehrspaltigen Listenfeldern oder Baumstrukturen. Mit dem Ereignis “Zelle bearbeiten” wird darauf reagiert, wenn der Benutzer in einspaltigen Listenfeldern, mehrspaltigen Listenfeldern oder Baumstrukturen eine Textänderung vornimmt.
- Damit beim Markieren eines Elements die gesamte Zeile markiert wird, klicken Sie mit der rechten Maustaste auf ein Listenfeld oder ein Baumstruktur-Element und wählen Sie aus dem Kontextmenü die Option **Ganze Zeile markieren** aus.
- Die Auswahl über die Tastatur in Listenfeldern und Baumstruktur-Elementen stimmt mit der in Ihrem Betriebssystem überein.



Hinweis Wenn die Tasten <Bild auf>, <Bild ab>, <Pos 1> oder <Ende> auf der Registerkarte **Tastenbelegung** des Eigenschaftens-Diaologfelds für ein Element festgelegt sind, können mit ihnen keine Elemente in Listenfeldern oder Baumstrukturen markiert werden.

Listenfelder

In LabVIEW 8.0 gab es folgende Verbesserungen und Veränderungen an Listenfeldern:

- Die Kontextmenü-Option **Sichtbare Objekte»Bildlaufleiste** bei einspaltigen Listenfeldern heißt nun **Vertikale Bildlaufleiste**. Die Eigenschaft “Bildlaufleiste sichtbar” heißt nun “Sichtbare Objekte:Vertikale Bildlaufleiste sichtbar”.
- Klicken Sie zum Anzeigen der Spaltenüberschrift mit der rechten Maustaste auf ein einspaltiges Listenfeld und wählen Sie im Kontextmenü die Option **Sichtbare Objekte»Spaltentitel** aus. Programmatisch wird die Spaltenüberschrift mit der Eigenschaft “Sichtbare Objekte:Spaltentitel sichtbar” ein- und ausgeblendet.
- Klicken Sie mit der rechten Maustaste auf ein einspaltiges Listenfeld und wählen Sie aus dem Kontextmenü die Option **Zeile davor einfügen** aus, um vor der ausgewählten Zeile eine Zeile einzufügen.
- Klicken Sie mit der rechten Maustaste auf ein einspaltiges Listenfeld und wählen Sie aus dem Kontextmenü die Option **Zeilenhöhe automatisch anpassen** aus, um die Zeilen so zu konfigurieren, dass sich die Zeilenhöhe automatisch entsprechend den Änderungen der Schriftart und mehrzeiligen Texteinträgen anpasst. Zur programmatischen Höhenänderung der Zeilen gibt es die Eigenschaft “Zeilenhöhe automatisch anpassen”.

- Für die mehrzeilige Eingabe in Zellen klicken Sie mit der rechten Maustaste auf ein einspaltiges Listenfeld und wählen Sie aus dem Kontextmenü die Option **Mehrzeilige Eingabe** aus. Programmatisch wird die mehrzeilige Eingabe in Zellen mit der Eigenschaft “Mehrfache Zeilenangabe” ermöglicht.
- Um pixelweise durch das Listenfeld zu scrollen, klicken Sie mit der rechten Maustaste auf ein mehrspaltiges Listenfeld und wählen Sie aus dem Kontextmenü die Option **Kontinuierlicher Bildlauf** aus. Zum programmatischen pixelweisen Scrollen im Listenfeld gibt es die Eigenschaft “Kontinuierlicher horizontaler Bildlauf”.

Baumstruktur-Elemente

In LabVIEW 8.0 gab es folgende Verbesserungen und Veränderungen an Baumstruktur-Elementen:

- Das Öffnen und Schließen von Elementen in der Baumstruktur ist mit der “+”- und “-”-Taste auf dem Zahlenblock möglich. Sie können jedoch auch das Baumstruktur-Element mit der rechten Maustaste anklicken und aus dem Kontextmenü die Optionen **Alle Elemente öffnen** bzw. **Alle Elemente schließen** wählen, um alle Elemente in der Baumstruktur zu öffnen bzw. zu schließen.
- Per Voreinstellung können immer nur solche Elemente gleichzeitig geöffnet werden, die demselben Element untergeordnet sind. Um die Auswahl von Elementen unter verschiedenen Elementen zu ermöglichen, klicken Sie mit der rechten Maustaste auf die Baumstruktur und deaktivieren Sie im Kontextmenü die Option **Auswahlmodus» Mehrfachauswahl nur auf gleicher Ebene**. Zur programmatischen Begrenzung der Auswahl mehrerer Elemente auf eine Ebene und zur Aufhebung dieser Beschränkung wird die Eigenschaft “Mehrfachauswahl auf gleicher Ebene” verwendet.



Hinweis Wenn die Option **Mehrfachauswahl nur auf gleicher Ebene** bei einem Baumstruktur-Element aktiviert ist, aber bereits mehrere Elemente auf verschiedenen Ebenen ausgewählt sind, kann sich das Element unvorhersehbar verhalten. Wenn die Auswahl auf einen gültigen Wert zurückgesetzt wird, funktioniert die Auswahl wieder normal.

- Um das Symbol zum Erweitern und Reduzieren auf der höchsten Hierarchie-Ebene auszublenden, klicken Sie mit der rechten Maustaste auf ein Baumstruktur-Element und entfernen Sie das Häkchen am Menüpunkt **Symbol für Erweitern/Reduzieren»Auf nullter Ebene anzeigen**. Die Ebene 0 wird dann linksbündig im Baumstruktur-Element angezeigt. Trotz Deaktivierung dieser Option können Sie ein Element der höchsten Hierarchieebene weiterhin durch einen Doppelklick erweitern bzw. reduzieren. Zum Deaktivieren dieser Funktion verbinden Sie den Parameter **Verwerfen?** des Ereignisses “Objekt

schließen?“ mit einer TRUE-Konstante. Um das Symbol zum Erweitern und Reduzieren der Ansicht auf der Ebene 0 programmatisch ein- bzw. auszublenden, wird die Eigenschaft “Symbol für Erweitern/Reduzieren:Für höchste Ebene anzeigen” verwendet.

Tabellen

Ist die horizontale Bildlaufleiste einer Tabelle nicht sichtbar, können Sie während der VI-Ausführung nicht mit Hilfe der Tastenkombination <Shift-Pfeil nach rechts> nach rechts scrollen. Bei ausgeblendeter vertikaler Bildlaufleiste können Sie während der VI-Ausführung nicht mit den Tasten <Shift-Pfeil nach unten> nach unten scrollen. Sie können jedoch mit diesen Tasten nach oben bzw. nach links scrollen, selbst wenn die Bildlaufleisten nicht sichtbar sind.

Verschieben von Elementen in String- und Baumstruktur-Elementen mit der Maus

Daten von String- und Baumstruktur-Elementen können in LabVIEW 8.0 per “Drag and Drop” mit der Maus verschoben werden. Zum Verschieben von Elementen aus einem String-Bedienelement markieren Sie den Text mit dem Bedienwerkzeug und verschieben Sie den markierten Text in ein kompatibles Bedienelement, beispielsweise ein anderes String-Element oder ein Baumstruktur-Element. Zum Verschieben von Elementen aus einer Baumstruktur markieren Sie das Element mit dem Bedienwerkzeug und verschieben Sie es in ein kompatibles Frontpanel-Element, beispielsweise ein anderes Baumstruktur-Element oder ein String-Element.

Einstellungen zum Verschieben des Inhalts von Bedienelementen sind mit Kontextmenüpunkten, Ereignissen, Eigenschaften und Methoden möglich. Mit der Funktion “Verschobene Daten ermitteln” können die verschobenen Daten einer Drag-and-Drop-Operation ausgegeben werden.

Deaktivieren des Zeilenumbruchs bei Text- und Anzeigeelementen

Klicken Sie zur Deaktivierung des Zeilenumbruchs bei String-Elementen mit der rechten Maustaste auf den Text und entfernen Sie das Häkchen neben dem Kontextmenüpunkt **Zeilenumbruch aktivieren**. Auf diese Weise wird der automatische Zeilenumbruch am Ende einer Zeile deaktiviert. Das String-Element muss sich zur Deaktivierung des Zeilenumbruchs im Modus “Normale Anzeige” befinden. Nach der Deaktivierung des Zeilenumbruchs können Sie eine horizontale Bildlaufleiste anzeigen. Klicken Sie mit der rechten Maustaste auf das String-Element und wählen Sie **Sichtbare Elemente»Horizontale Bildlaufleiste** aus dem Kontextmenü.

Programmatisch werden Zeilenumbrüche mit der Eigenschaft “Zeilenumbruch aktivieren” deaktiviert. Um die horizontale Bildlaufleiste einzublenden, verwenden Sie die Eigenschaft “Horizontale Bildlaufleiste: Sichtbar”.

Pfad-Bedienelemente

Das bisherige Dialogfeld **Suchoptionen** wird durch die Seite **Suchoptionen** im Dialogfeld **Eigenschaften für Pfad** ersetzt. Zur Anzeige der Seite **Suchoptionen**, klicken Sie das Pfad-Bedienelement mit der rechten Maustaste an und wählen Sie **Suchoptionen**. Die Seite enthält folgende Neuerungen:

- Mit Hilfe des Textfelds **Schaltflächenbeschriftung** lässt sich die Schaltflächenbeschriftung ändern. Programmatisch wird die Schaltflächenbeschriftung mit der Eigenschaft “Suchoptionen: Schaltflächenbeschriftung” geändert.
- Aktivieren Sie die Option **LLBs wie Ordner behandeln**, um dem Benutzer die Auswahl einer Datei zu ermöglichen.
- Das Pulldown-Menü **Auswahlmodus** wurde durch folgende Optionen ersetzt: **Nur Dateien**, **Nur Ordner**, **Dateien oder Ordner**, **Nur existierende**, **Nur neue** und **Neue oder existierende**.

Bildlaufleistenelemente

Mit den horizontalen und vertikalen Bildlaufleistenelementen können Sie Elementen mit Daten auf einer größeren Fläche benutzerspezifische Bildlaufleisten hinzufügen. Um den Wert einer Bildlaufleiste zu ändern, verschieben Sie sie mit dem Bedienwerkzeug, klicken Sie auf die Pfeile oder klicken Sie auf die leere Fläche zwischen der Leiste und den Pfeilen.

Mit Hilfe der Eigenschaften “Dok Min”, “Dok Max”, “Inkrementieren”, “Seitengröße”, “Auf Mindestwert setzen” und “Auf Höchstwert setzen” können die Seitengröße und die Endwerte für die Bildlaufleiste programmatisch festgelegt werden.

Array-Bildlaufleisten

Die horizontale und die vertikale Bildlaufleiste für ein Array lassen sich durch Rechtsklick auf ein Bedien- oder Anzeigeelement auf dem Frontpanel und Auswahl der Option **Sichtbare Objekte»Vertikale Bildlaufleiste** bzw. **Sichtbare Objekte»Horizontale Bildlaufleiste** im Kontextmenü einblenden. Nur bei Arrays mit mehreren Dimensionen kann sowohl die horizontale als auch die vertikale Bildlaufleiste angezeigt werden. Zur Anzeige der vertikalen oder horizontalen Bildlaufleiste ziehen Sie ein 1D-Array in der entsprechenden Richtung auf.

Mit der Eigenschaft “Vertikale Bildlaufleiste sichtbar” bzw. “Horizontale Bildlaufleiste sichtbar” können Bildlaufleisten programmatisch eingeblendet werden.

Verschiedene Neuerungen und Änderungen

In LabVIEW 8.0 gibt es folgende Verbesserungen und Änderungen am Frontpanel:

- Zum Aktivieren bzw. Deaktivieren der Plotlegende während der VI-Ausführung klicken Sie einen Graphen oder ein Diagramm mit der rechten Maustaste an und wählen aus dem Kontextmenü die Option **Fortgeschritten»Plot-Legende deaktivieren** aus. Programmatisch wird die Plotlegende an einem Signalverlaufdiagramm, Signalverlaufgraphen oder Mischsignal-Graphen mit der Eigenschaft “Legende:Deaktivieren” ein- und ausgeblendet. Wenn Sie die Plotlegende deaktivieren, kann die Darstellung des Plots in einem Graphen oder Diagramm während der Ausführung nicht geändert werden.
- In LabVIEW 7.x und Vorgängerversionen wird an der y-Achse des digitalen Signalverlaufgraphen per Voreinstellung eine Achsenbeschriftung angezeigt. In LabVIEW 8.0 hat die y-Achse per Voreinstellung keine Beschriftung. Wenn die Achsenbeschriftung angezeigt wird, lautet die automatische Beschriftung jetzt **Digitale Plots** anstelle von **Amplitude**.
- In Listenfeldern, mehrspaltigen Listenfeldern, Tabellen, digitalen Bedienelementen und Baumstrukturen können Sie jetzt mit dem Mausrad nach oben und unten scrollen.
- Klicken Sie eine Komponente eines Bedien- oder Anzeigeelements mit der rechten Maustaste an, die Sie ändern möchten, und wählen Sie aus dem Kontextmenü die Option **Fortgeschritten»Anpassen** aus. Daraufhin öffnet sich der **Bedienelement-Editor** für die ausgewählte Komponente.
- Wenn Sie den Zeiger oder die Nadel von Schiebe- oder Drehreglern bewegen, wird der Wert des Anzeigeelements in einem Hinweisstreifen angezeigt. Dabei gelten das gleiche Format und die gleiche Genauigkeit wie bei der Zahlenanzeige des Elements. Wenn das Element mehrere Zahlenanzeigen hat, passt sich der Hinweisstreifen in Format und Genauigkeit der Zahlenanzeige an, die zum aktuell bedienten Schieberegler gehört. Zum Deaktivieren des Hinweisstreifens klicken Sie mit der rechten Maustaste auf das Bedienelement und wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus. Klicken Sie anschließend auf die Registerkarte **Erscheinungsbild** und entfernen Sie das Häkchen an der Option **Hinweisstreifen anzeigen**. Programmatisch werden die Hinweisstreifen für Werte mit der Eigenschaft “Wert-Hinweisstreifen anzeigen” deaktiviert.

- Numerische Bedien- und Anzeigeelemente arbeiten mit folgenden SI-Vorsätzen: Zenti-, Dezi-, Deko- und Hekto-.

LabVIEW-MathScript (Windows)

Mit Hilfe des **LabVIEW-MathScript-Fensters** können mathematische Befehle eingegeben und ausgeführt werden, mathematische Skripts erstellt sowie numerische und grafische Darstellungen einer Variablen angezeigt werden. Wählen Sie zur Anzeige dieses Fensters **Werkzeuge» MathScript-Fenster**. Im **LabVIEW MathScript-Fenster** werden alle nach Ausführung eines Befehls vorliegenden Ergebnisse und alle aufgerufenen Befehle angezeigt sowie alle Variablen, die Sie erstellt oder ausgewählt haben. Bei LabVIEW MathScript können fast alle Skripte ausgeführt werden, die in der MATLAB®-Syntax verfasst sind. Die MathScript-Umgebung unterstützt allerdings einige MATLAB-Funktionen nicht.



Hinweis MathScript gibt es nur im LabVIEW Professional Development System.

Sie können Funktionen und Skripte schreiben, die im **LabVIEW-MathScript-Fenster** oder dem **“MathScript-Knoten”** verwendet werden.

Mit dem MathScript-Knoten können Sie mathematische Formeln und Ausdrücke in einer MATLAB-ähnlichen Syntax auf dem Blockdiagramm berechnen. Sie können Skripte speichern, die Sie im **LabVIEW-MathScript-Fenster** erstellt haben, und diese dann im MathScript-Knoten laden. Sie können Skripte, die im MathScript-Knoten erstellt wurden, auch speichern und in das **LabVIEW MathScript-Fenster** laden.

Matrix-Datentyp

Mit Hilfe des Matrix-Datentyps gestaltet sich das Erstellen von Modellen für mathematische Probleme einfacher und produktiver. Alle vorhandenen polymorphen VIs und Funktionen für Matrix-Operationen erkennen den Matrix-Datentyp und führen entsprechende matrizenspezifische Algorithmen durch. Mathematik-VIs und -Funktionen, die für komplexe Matrix-Operationen verwendet werden, arbeiten mit Matrizen am Eingang und geben Matrizen aus.



Hinweis In LabVIEW 7.x und Vorgängerversionen sind die VIs “e hoch Matrix”, “Logarithmus einer Matrix”, “Potenzierung der Matrix” und “Quadratwurzel aus Matrix” nur im LabVIEW Full and Professional Development System enthalten. In LabVIEW 8.0 gibt es diese VIs auch im Base Package.

Verwenden Sie zum Darstellen von Matrizen lieber den Matrix-Datentyp anstatt eines 2D-Arrays, da der Matrix-Datentyp Zeilen und Spalten von reellen oder komplexen Skalarwerten für Matrix-Operationen speichert, insbesondere für einige Operationen der linearen Algebra. Die Mathematik-VIs und -Funktionen, die Matrixoperationen durchführen, arbeiten mit Matrizen und geben auch Matrizen als Ergebnisse an nachfolgende polymorphe VIs und Funktionen weiter. Wenn ein Mathematik-VI oder eine Mathematik-Funktion keine Matrix-Operation durchführt, aber mit Matrizen arbeitet, wird die Matrix automatisch in ein 2D-Array umgewandelt. Wenn Sie ein 2D-Array mit einer Funktion oder einem VI verbinden, das per Voreinstellung Matrix-Operationen durchführt, wandelt das VI bzw. die Funktion das 2D-Array je nach dem Datentyp des 2D-Arrays automatisch in eine reelle oder komplexe Matrix um.

Die meisten numerischen Funktionen sind sowohl für den Matrix-Datentyp als auch für Matrix-Operationen geeignet. So kann zum Beispiel eine Matrix mit der Funktion "Multiplizieren" mit einer anderen Matrix oder einer Zahl multipliziert werden. Sie können grundlegende numerische Datentypen und komplexe Funktionen der linearen Algebra kombinieren, um Algorithmen für exakte Matrix-Operationen zu erstellen.

In LabVIEW 7.x und Vorgängerversionen waren mathematische Operationen an Matrizen nur mit den VIs zur linearen Algebra möglich. Obwohl ein 2D-Array Matrix-Daten speichern kann, wird das Array für einige numerische Funktionen (zum Beispiel einige Funktionen der linearen Algebra) anders behandelt als Matrix-Daten in einem Matrix-Bedienelement. In LabVIEW 7.x und Vorgängerversionen kann ein 2D-Array nicht von einer Matrix unterschieden werden. Daher müssen Sie die Dimensionsgröße des Arrays und die Array-Elemente vor dem Durchführen der mathematischen Funktion überprüfen, um sicherzugehen, dass die Matrix-Daten für eine Matrix-Operation zulässig sind. Beim Matrix-Datentyp brauchen weder Dimensionsgröße eines Arrays überprüft noch die Array-Elemente analysiert zu werden. Wenn sich die Dimensionen der beiden Matrizen unterscheiden, die am Eingang anliegen (z. B. eine Matrix mit 2×3 und eine weitere mit 4×4), gibt das VI bzw. die Funktion eine leere Matrix aus.



Hinweis Wenn das VI bzw. die Funktion Daten in oder von einer Matrix bzw. von einem 2D-Array umwandelt, werden Typumwandlungspunkte angezeigt. Diese Art der Umwandlung hat keinen Einfluss auf die Ausführungsgeschwindigkeit, da die Matrizen auf die gleiche Art und Weise gespeichert werden wie 2D-Arrays.

Auf dem Blockdiagramm sieht der Matrix-Datentyp wie ein reelles 2D-Array oder ein komplexes 2D-Array mit einem anderen Verbindungsmuster aus. Die VIs und Funktionen, die mit dem Matrix-Datentyp arbeiten, führen auch matrizenspezifische Operationen aus, wenn Sie einen Matrix-Datentyp am Eingang anlegen.

Wenn Sie einen Matrix-Datentyp mit einer der folgenden Funktionen verbinden, wird diese durch ein VI ersetzt, dessen SubVIs mit dem Matrix-Datentyp arbeiten:

- Funktion “Gleich?”
- Funktion “Ungleich?”
- Funktion “Absoluter Wert”
- Funktion “Addieren”
- Funktion “Multiplizieren”
- Funktion “Quadratwurzel”
- Funktion “Subtrahieren”
- Funktion “Exponentialfunktionen”
- Funktion “Natürlicher Logarithmus”
- Funktion “X hoch Y”
- Funktion “Re/Im nach Komplex”
- Funktion “Komplex nach Re/Im”
- Funktion “Polar nach Komplex”
- Funktion “Komplex nach Polar”

Das resultierende VI hat das gleiche Symbol, enthält aber einen matrizen-spezifischen Algorithmus. Wenn Sie die Matrizen von den Eingängen trennen, bleibt der Knoten weiterhin ein VI. Zum Wiederherstellen der Originalfunktion müssen am Eingang andere Datentypen anliegen. Wenn Sie einen Datentyp mit einer Funktion verbinden und aufgrund dieses Datentyps eine grundlegende mathematische Operation nicht funktioniert, gibt die Funktion eine leere Matrix oder **NaN** aus. Wenn Sie zum Beispiel eine Matrix mit einer Dimension von 2×3 mit einem Eingang der Funktion “Multiplizieren” und eine weitere Matrix (4×4) mit einem anderen Eingang verbinden, gibt die Funktion eine leere Matrix aus.



Hinweis Wenn Sie ein VI von LabVIEW 7.x in LabVIEW 8.0 öffnen und darin Mathematik-VIs enthalten und mit Funktionen verbunden sind, die den Matrix-Datentyp verwenden können, werden die Typdefinitionen deaktiviert und das Verhalten von LabVIEW 7.x beibehalten. Um anzuzeigen, dass sich diese Funktion wie in LabVIEW 7.x verhält, wird an der Funktion ein rotes 7.x-Symbol eingeblendet.

In LabVIEW 7.0 und Vorgängerversionen werden nur 1D-Arrays im Speicher abgeglichen. Dadurch werden Array-Operationen beschleunigt. In LabVIEW 7.1 und späteren Versionen werden 1D- und 2D-Arrays im Speicher abgeglichen. Das ist bei der Durchführung von Operationen der linearen Algebra und Operationen hilfreich, in denen der Matrix-Datentyp vorkommt.

Verbessertes Dialogfeld “Optionen”

Wählen Sie **Werkzeuge»Optionen**, um zum Dialogfeld **Optionen** zu gelangen. Um zwischen den Seiten des Dialogfelds **Optionen** zu wechseln, wählen Sie die gewünschte Option aus der Liste **Kategorie** aus. Das Dialogfeld **Optionen** wurde in folgender Hinsicht verbessert:

- Die Seite **Verschiedenes** wurde in **Umgebung** umbenannt. Die Option **Zulässige Rückgängig-Schritte pro VI** wurde von der Seite **Blockdiagramm** auf die Seite **Umgebung** verschoben.
- Die Seite **VI Server: TCP/IP-Zugriff** wurde in **VI-Server: Computer-Zugriff** umbenannt. Auch die Namen der Optionen auf der Seite wurden geändert.
- Auf der Seite **VI-Server: Benutzerzugriff** können Sie nun festlegen, welche Benutzer via VI-Server mit bestimmten VIs arbeiten können.
- Auf der Seite **Menü-Verknüpfungen** können nun Tastenkombinationen für LabVIEW-Menüpunkte festgelegt werden.
- Auf der Seite **Versionsverwaltung** (nur LabVIEW Professional Development System) können Sie einen Versionsverwaltungsprovider einrichten und Einstellungen zur Versionsverwaltung in LabVIEW vornehmen.
- Unter **Sicherheit** befinden sich Sicherheitsoptionen für die Nutzung von LabVIEW.
- Auf der Seite **Engine für Umgebungsvariablen** können Sie Einstellungen für die Synchronisierung von Umgebungsvariablen vornehmen.
- Mit Hilfe der Optionen **Beschreibungen rasten bei Elementen/Terminals an voreingestellten Positionen ein**, **Beschriftungen standardmäßig gesperrt** und **Standard-Beschriftungsposition** auf den Seiten **Frontpanel** und **Blockdiagramm** können Sie die Positionen von Beschriftungen und Untertitel für neue Objekte festlegen.
- Damit die Palettenobjekte in Form einer Baumstruktur angezeigt werden, wählen Sie die Option **Baumstruktur** im Pulldown-Menü **Format** auf der Seite **Paletten “Elemente” und “Funktionen”**. Aktivieren Sie die Option **Palettenobjekte sortieren**, um Palettenobjekte in alphabetischer Reihenfolge zu sortieren, wenn Sie die Option **Text** oder **Baumstruktur** aus dem Pulldown-Menü **Format** ausgewählt haben.
- Mit Hilfe der Option **Name des Dienstes** auf der Seite **VI-Server: Konfiguration** können Sie den Namen des TCP/IP-Dienstes für Verbindungsanfragen festlegen.

Dialogfelder “Speichern unter” und “Für vorige Version speichern”

Wählen Sie **Datei»Speichern unter** für ein zuvor gespeichertes VI, um zum Dialogfeld **Speichern unter** zu gelangen. Wählen Sie **Datei»Für vorige Version speichern**, um zum Dialogfeld **Für vorige Version speichern** zu gelangen.

Das Dialogfeld **Mit Optionen speichern** wurde entfernt. Verwenden Sie stattdessen das Dialogfeld **Speichern unter** bzw. **Für vorige Version speichern**. Sie können auch das Dialogfeld **Quellcodepaket-Einstellungen** im Feld **Speichern unter** öffnen, um ein Quellcodepaket zu erstellen und Einstellungen für angegebene VIs zu konfigurieren, wie Hinzufügen von Passwörtern oder Entfernen von Blockdiagrammen.

Verbesserungen bezüglich der Dateigröße

Wenn Sie VIs früherer Versionen in LabVIEW 8.0 speichern, erhöht sich deren Dateigröße erheblich. Die Dateigröße von in LabVIEW 8.0 gespeicherten VIs liegt ca. 55% unter der einer vergleichbaren Datei in LabVIEW 7.1. Die Dateigröße von LLBs hat sich gegenüber LabVIEW 7.1 um ungefähr 20% verringert.

Verwendung von DLLs in mehreren LabVIEW-Versionen

Wenn Sie in LabVIEW 7.x und Vorgängerversionen eine DLL in einer LabVIEW-Version erstellen und die DLL dann in einer anderen LabVIEW-Version öffnen, wird möglicherweise ein Fehler ausgegeben, da die DLL von einer anderen LabVIEW-Runtime-Engine aufgerufen wird. In LabVIEW 8.0 können DLLs in mehreren LabVIEW-Versionen bzw. in mehreren Versionen der Runtime-Engine genutzt werden, wenn Sie beim Erstellen der DLL `labviewv.lib` statt `labview.lib` verwenden.

Festlegen der Fensterposition bei Ausführung

Um die Größe und die Position des Frontpanel-Fensters festzulegen, wählen Sie **Datei»VI-Einstellungen** und klicken Sie im Pulldown-Menü **Kategorie** auf **Fensterposition bei Ausführung**. Programmatisch werden Position und Größe des Frontpanels mit den Methoden unter Frontpanel:Laufzeitposition geändert.

Die Option **Frontpanel bildschirmfüllend anzeigen** bisheriger LabVIEW-Versionen, die sich auf der Seite **Fenstergröße** im Dialogfeld **VI-Einstellungen** befindet, wurde durch den Menüpunkt **Maximiert** im Pulldown-Menü **Position** ersetzt.

Analog dazu wurde die Option **Automatisch zentrieren** im Dialogfeld **Fenstererscheinungsbild anpassen** durch den Menüpunkt **Zentriert** im Pulldown-Menü **Position** ersetzt.

Wenn Sie in älteren Versionen von LabVIEW für ein VI die Optionen **Automatisch zentrieren** oder **Frontpanel bildschirmfüllend anzeigen** ausgewählt haben, ist das VI beim Anhalten nicht auf die Position des Bearbeitungsmodus zurückgesprungen. Das VI blieb zentriert oder bildschirmfüllend. In LabVIEW 8.0 springt das VI beim Anhalten zurück auf die Position des Bearbeitungsmodus.

Einstellen von Sprachen des Laufzeitmenüs

Spracheinstellungen für ausführbare Anwendungen können auf eine der nachfolgenden Arten vorgenommen werden. Die zuerst aufgelisteten Methoden haben eine höhere Priorität als die nachfolgenden.

- Führen Sie die Anwendung mit der Kommandozeilenoption `-lang language` aus, wobei *language* für die englische Bezeichnung der zu verwendenden Sprache steht.
- Fügen Sie der `*.ini`-Datei für die Anwendung folgende Zeile hinzu: `AppLanguage=language`, wobei *language* für die englische Bezeichnung der zu verwendenden Sprache steht.
- **(Windows)** Wenn Sie keine Sprache in der Kommandozeile oder in der `*.ini`-Datei der Anwendung festlegen, hat die Anwendung die gleiche Sprache wie das Betriebssystem.
- Unterstützt die Anwendung die Sprache des Betriebssystems nicht, wird in der Anwendung die Standardsprache eingestellt.
- Wenn Sie keine der oben aufgeführten Spracheinstellungen festlegen, wird in der Anwendung die erste zur Verfügung stehende, unterstützte Sprache bei alphabetischer Reihenfolge verwendet. Wenn keine der unterstützten Sprachen vorliegt, kann die Anwendung nicht geladen werden.

Verbesserungen am VI-Server

Der VI-Server liefert detailliertere Fehlerprotokolle als in LabVIEW 7.x und Vorgängerversionen. Wenn zum Beispiel bei der Funktion “VI-Referenz” ein Fehler auftritt, dann gibt das Fehlerprotokoll an, welches VI die Fehlerquelle ist. Wenn Sie ein VI ohne Frontpanel laden, erhalten Sie eine Fehlermeldung, in der steht, dass LabVIEW das Frontpanel nicht laden kann. Fehlermeldungen enthalten nun auch die Namen von bestimmten Eigenschaften und Methoden, die Fehler erzeugen.

Verwendung von VI-Server-Referenzen

Das Element "Referenzkonstante" wurde in VI-Server-Referenz umbenannt. Sie können diese Refnum in das Blockdiagramm einfügen und sie mit dem aktuellen VI, einer Anwendung oder einem Element im VI verknüpfen. Mit Hilfe dieser Refnum können Sie Eigenschaften und Methoden auf ein VI, eine Anwendung oder Bedien- bzw. Anzeigelemente anwenden.

Um die Refnum mit dem aktuellen VI bzw. mit der aktuellen Anwendung zu verknüpfen, klicken Sie sie auf dem Blockdiagramm mit dem Bedienwerkzeug an und wählen Sie **Diese Anwendung** oder **Dieses VI**. Stattdessen können Sie die Refnum auch mit der rechten Maustaste anklicken und **Verknüpfen mit»Diese Anwendung** oder **Verknüpfen mit»Dieses VI** auswählen. Um die Refnum einem Bedien- oder Anzeigelement innerhalb des VIs zuzuordnen, klicken Sie die Refnum mit der linken oder rechten Maustaste an und wählen Sie dann aus dem Kontextmenü das gewünschte Element aus.

Von LabVIEW erstellte Anwendungen können keine VI-Server-Referenzen für polymorphe VIs erzeugen, sondern das polymorphe VI wird entfernt und stattdessen werden die polymorphen Instanzen erstellt. VI-Server-Operationen werden anschließend direkt an den polymorphen Instanzen durchgeführt.

Sicherheit für VI-Server

Der VI-Server enthält eine Sicherheitsfunktion. Damit haben Sie mehr Kontrolle darüber, welche Client-Rechner eine Verbindung mit dem VI-Server herstellen können. In LabVIEW 7.x und Vorgängerversionen werden die Zugriffsrechte je nach der Adresse des Clients gewährt oder verweigert. In LabVIEW 8.0 kann ein VI-Server das Erstellen von Verbindungen je nach Benutzern und Gruppen gewähren oder verweigern.

Die NI-Sicherheitsfunktion für Server ist per Voreinstellung deaktiviert. In diesem Fall können Verbindungen nur auf Grundlage der Computer-Adresse gewährt oder verweigert werden. Zur Aktivierung der NI-Sicherheitsfunktion müssen Sie festlegen, ob ein Benutzer oder eine Gruppe Zugriff erhalten soll oder nicht. Zur Seite **VI-Server: Benutzerzugriff**, auf der Sie die Zugriffsrechte festlegen können, gelangen Sie folgendermaßen: Wählen Sie **Werkzeuge»Optionen**, um zum Dialogfeld **Optionen** zu gelangen und wählen Sie unter **Kategorie** die Option **VI-Server: Benutzerzugriff** aus. Ist die Zugriffsliste leer, ist die Sicherheitsfunktion für den VI-Server deaktiviert.

In LabVIEW 8.0 können Clients Verbindungen zu VI-Servern auf die gleiche Weise erstellen wie in LabVIEW 7.x und Vorgängerversionen. Gibt es auf dem Server VI-Server-Beschränkungen für Benutzer und Gruppen,

muss sich der Client möglicherweise anmelden, bevor die Verbindung erstellt wird. Dazu gibt es entweder das Dialogfeld **NI-Sicherheit: Anmelden** oder die Methode “NI-Sicherheit: Anmelden” für programmatisches Anmelden. Ist ein Benutzer nicht berechtigt, eine Verbindung zu erstellen, beendet der Server die Verbindung und gibt an den Client den Fehler (1379) `kLVE_NISecurity_AuthenticationFailed` aus.

Wenn ein Client wechselt, während eine Verbindung mit dem VI-Server besteht, sendet der Client automatisch eine Nachricht an den Server, um ihn über den Benutzerwechsel zu informieren. Erhält der neue Client keine Berechtigung für eine Verbindung, beendet der Server die Verbindung und gibt die Fehlermeldung (1379) `kLVE_NISecurity_AuthenticationFailed` aus.

Klassen-Browser-Fenster

Im Fenster **Klassen-Browser** können Sie eine Objektbibliothek auswählen und sich Klassen, Eigenschaften und Methoden innerhalb der Bibliothek anzeigen lassen. Mit der ausgewählten Eigenschaft oder Methode können Sie Eigenschafts- oder Methodenknotten erstellen. Im **Klassen-Browser** können Sie auch nach Klassen, Eigenschaften und Methoden suchen sowie Eigenschaften und Methoden in Punktnotation erstellen. Zum Öffnen des Fensters **Klassenbrowser** klicken Sie auf **Ansicht»Klassenbrowser**.

Zum Bearbeiten eines Eigenschafts- oder Methodenknottens klicken Sie die gewünschte Eigenschaft oder Methode im Knoten mit der rechten Maustaste an und wählen Sie aus dem Kontextmenü die Option **Eigenschaft auswählen** bzw. **Methode auswählen** aus.

Verwenden von .NET-Assemblies

LabVIEW lädt automatisch die neueste Version der .NET Common Language Runtime (CLR). Wenn Sie eine .NET-Applikation mit einer älteren Version von .NET erstellen, installieren Sie eine neuere Version von .NET und erzeugen Sie die Applikation erneut. Benutzer einer älteren .NET-Version können Ihre Applikationen nicht mehr ausführen. Mit Hilfe einer Konfigurationsdatei können Sie eine spezielle CLR-Version laden. Auf diese Weise können Sie eine neuere .NET-Version installieren und weiterhin .NET-Applikationen entwickeln und andere Benutzer können ihre Applikationen nach wie vor mit der älteren .NET-Version nutzen.

In LabVIEW 7.x müssen Sie die Option **Werkzeuge»Fortgeschritten» .NET-Assembly-Referenzen** auswählen, um private .NET-Assembly-Dateien manuell zu registrieren, so dass Sie mit VIs arbeiten können, die sich auf diese Assemblys beziehen. In LabVIEW 8.0 müssen private .NET-Assembly-Dateien nicht mehr registriert werden und der Menüpunkt **Werkzeuge»Fortgeschritten» .NET-Assembly-Referenzen** wurde ent-

fernt. Eine Refnum für eine private .NET-Assembly wird jetzt mit einem Konstruktor-Knoten erzeugt. LabVIEW speichert den relativen Pfad von der Assembly zu dem VI, in dem das .NET-Objekt enthalten ist. Am schnellsten werden Assemblys geladen, wenn sie im Global Assembly Cache (GAC) im selben Verzeichnis wie die LabVIEW-Projektdatei oder im Unterverzeichnis des LabVIEW-Projekts gespeichert werden.

Die .NET-CLR bestimmt anhand von Konfigurationseinstellungen, welche Version der Assembly geladen werden muss. Durch diese Einstellungen kann die von Ihnen gewünschte Version unberücksichtigt bleiben. So kann der Systemadministrator beispielsweise den Computer so einrichten, dass immer anstelle von 1.0.0.0 die Version 1.0.0.1 geladen wird. Wenn Sie dann versuchen, die Version 1.0.0.0 zu laden, wird die Version automatisch auf 1.0.0.1 erhöht. Außerdem öffnet sich ein Dialogfeld, in dem Sie über die Änderung der Version informiert werden.

Wenn Sie ein VI, das eine private Assembly verwendet, in einen anderen Ordner oder auf einen anderen Rechner verschieben, muss die Assembly am selben Speicherort relativ zum VI verbleiben. Wird keine Assembly-Datei im GAC, im Verzeichnis, in dem das Projekt enthalten ist, oder im relativen Pfad, der im VI gespeichert ist, gefunden, werden Sie aufgefordert, die Assembly-Datei selbst zu suchen.

Wenn Sie aus einem VI mit einer privaten Assembly eine DLL oder ausführbare Anwendung erzeugen, kopiert LabVIEW die benötigten Assembly-Dateien in das Unterverzeichnis `Daten` des Programm- oder Bibliotheksverzeichnisses.

In LabVIEW 8.0 beziehen sich `*.config`-Dateien auf ein gespeichertes Projekt, eine DLL oder eine Anwendung.

Suchmaschine für Gerätetreiber

Mit Hilfe der NI-Suchmaschine für Gerätetreiber (**Werkzeuge»Instrumentierung»Instrumententreiber suchen**) lassen sich LabVIEW-Plug-and-Play-Gerätetreiber einfach suchen und installieren. Die LabVIEW-Entwicklungsumgebung muss dazu nicht verlassen werden. Die Suchmaschine für Gerätetreiber zeigt auch die Gerätetreiber unter `instr.lib` an und sucht nach angeschlossenen Geräten mit Hilfe von NI-VISA.



Hinweis Die Suchmaschine für Gerätetreiber ist unter Mac OS nicht verfügbar.

Assistent für Gerätetreiber-VIs

Mit dem Assistenten für Gerätetreiber-VIs wird die vorhandene Gerätetreiber-Projektbibliothek erweitert. Erstellen oder öffnen Sie ein Gerätetreiber-Projekt, klicken Sie mit der rechten Maustaste auf die Projektbibliothek im Fenster **Projekt-Explorer** und wählen Sie aus dem Kontextmenü die Option **Neu»Gerätetreiber-VI** aus, um den Assistenten für Instrumententreiber-VIs zu starten. Mit Hilfe des Assistenten können Sie ein Gerätetreiber-VI erstellen und dieses VI in eine Gerätetreiber-Projektbibliothek einfügen. Mit dem Assistenten können Sie auch Parameter eines Instrumenten-Befehlsstrings konfigurieren, die auf Frontpanel-Bedienelemente reagieren und festlegen, wie die Antwort des Geräts ausgewertet werden soll.

Assistent für Gerätetreiberprojekte

Durch den Assistenten für Gerätetreiberprojekte (aufrufbar über **Werkzeuge»Instrumentierung»Gerätetreiberprojekt erstellen**) wird das Erstellen eines Gerätetreiber-Projekts automatisiert. Das gilt auch für VIs zur Steuerung programmierbarer Hardware (wie GPIB-, RS-232-, Ethernet- oder USB-Hardware). Mit Hilfe des Assistenten können Gerätetreiber-VIs und Palettenmenüs zum Anzeigen der Gerätetreiber auf der **Funktionen**-Palette erstellt werden. Sie können einen Gerätetreiber von einer Klassenvorlage oder einer allgemeinen Vorlage für den Nachrichtenaustausch erstellen oder bei der Entwicklung von einem vorhandenen Treiber ausgehen.

Die zur Verfügung stehenden Klassenvorlagen lauten: “Gleichspannungsversorgung”, “Digital-Multimeter”, “Funktionsgenerator” und “Oszilloskop”. Die allgemeine Vorlage eignet sich für jedes beliebige Gerät, das Nachrichten senden und empfangen kann. Ist das Gerät mit einem vorhandenen Gerät vergleichbar, können Sie einen neuen Treiber auf Grundlage des vorhandenen erstellen.

Wenn Sie einen neuen Gerätetreiber auf Grundlage eines vorhandenen erstellen, muss sich das dazugehörige Projekt im Verzeichnis `labview\instr.lib` befinden, bevor der Assistent für Gerätetreiber-Projekte gestartet werden kann. Der vorhandene Treiber muss ein Treiber im Projekt-Stil sein. Sie können solche Treiber in der Suchmaschine für NI-Gerätetreiber herunterladen und installieren.

Der Assistent für Gerätetreiber-Projekte erstellt ein LabVIEW-Projekt im Standardformat für LabVIEW-Plug-and-Play-Gerätetreiber. Das Projekt enthält VIs und Palettenmenü-Dateien, die den Richtlinien für Gerätetreiber entsprechen. Wenn der Assistent das Projekt erstellt hat, muss jedes VI so angepasst werden, dass es mit einem Gerät zusammenarbeitet. Mit Hilfe des Assistenten für Gerätetreiber-VIs können Sie ein Gerätetreiber-VI erstellen und dieses VI in eine Gerätetreiber-Projektbibliothek einfügen.

Verwenden von ablaufinvarianten VIs und Fehlersuche

Folgende Arbeitsschritte sind in LabVIEW an ablaufinvarianten VIs möglich:

- Ablaufinvariante VIs lassen sich auf Fehler untersuchen. Öffnen Sie dazu eine Kopie des Blockdiagramms eines beliebigen ablaufinvarianten VIs. Diese Kopie des Blockdiagramms kann jedoch nicht bearbeitet werden. Im Blockdiagramm können Sie während der Ausführung Haltepunkte setzen, Sonden abfragen und die Highlightfunktion sowie die Einzelschrittausführung aktivieren. Zum Ausführen der Fehlersuche in einem ablaufinvarianten VI wählen Sie **Datei»VI-Einstellungen**, um zum Dialogfeld **VI-Einstellungen** zu gelangen, wählen Sie **Ausführung** aus und aktivieren Sie die Option **Fehlerbehandlung aktiviert**.
- Im Dialogfeld **VI-Einstellungen** können Sie ein ablaufinvariantes VI so einstellen, dass das Frontpanel während der Ausführung geöffnet und wahlweise nach der VI-Ausführung wieder geschlossen wird.
- Mit Hilfe des VI-Servers können die Bedien- und Anzeigeelemente des Frontpanels eines ablaufinvarianten VIs während der VI-Ausführung programmatisch gesteuert werden. Die Bedien- und Anzeigeelemente können jedoch nicht bearbeitet werden. Durch verschiedene VI-Server-Methoden und -Eigenschaften, wie “Bedienelementwert:Lesen [Serialisiert]”, “Bedienelementwert:Setzen [Serialisiert]” und “Ausführen” können Sie mit der ablaufinvarianten Kopie eines VIs kommunizieren, wenn es eine Refnum für diese Kopie gibt.
- Mit dem VI-Server lässt sich während der Ausführung eine Kopie des Frontpanels eines ablaufinvarianten VIs erstellen. Verwenden Sie dazu die Funktion “VI-Referenz öffnen”, um eine VI-Server-Refnum zu erhalten. Verbinden Sie eine strikt-typisierte VI-Refnum mit dem Eingang **Typ-Bezeichner** oder verbinden Sie **offen für ablaufinvariante Ausführung** mit dem Eingang **Option**. Beim Erstellen der Refnum wird automatisch eine Kopie des VIs erstellt. Das Frontpanel eines ablaufinvarianten VIs kann aber auch mit dem VI-Server oder dem Dialogfeld **VI-Einstellungen** geöffnet werden.
- Sie können einen Case einer Ereignisstruktur so konfigurieren, dass er Ereignisse für ein Frontpanelobjekt eines ablaufinvarianten VIs bearbeitet.
- Das Frontpanel eines ablaufinvarianten VIs kann über ein Netzwerk entweder von LabVIEW aus oder über einen Webbrowser betrachtet und bedient werden. Dazu ist eine Verbindung via Web-Server mit dem betreffenden Rechner erforderlich. Beim Web-Server können mehrere Benutzer Zugriff auf die Controller-Schnittstelle eines VIs haben. Für jede Verbindung gibt es eine eigene Kopie.

- Das Fassaden-VI eines XControls muss ablaufinvariant sein.
- Das Frontpanel eines ablaufinvarianten VIs kann auch ein Unterpanel sein.
- In früheren LabVIEW-Versionen wird für alle Kopien des ablaufinvarianten VIs nur ein Frontpanel angezeigt, das heißt, die Werte der Bedien- und Anzeigeelemente waren nicht immer ganz genau. In LabVIEW 8.0 hat jede Kopie eines ablaufinvarianten VIs ihr eigenes Frontpanel. Auf diese Weise erhalten Sie immer genaue Werte für Bedien- und Anzeigeelemente.

Wenn Sie ein ablaufinvariantes SubVI vom Blockdiagramm öffnen, wird anstelle des Original-VIs eine Kopie des VIs geöffnet. In der Titelleiste des VIs steht *(Kopie)*, um anzuzeigen, dass es sich bei dem VI um eine Kopie handelt. Die VI-Kopie kann nicht bearbeitet werden.

Erstellen von Sonden nach Start eines VIs

Sie können ein VI so konfigurieren, dass es die Verbindungswerte erhält, so dass beim Erzeugen einer Sonde im Blockdiagramm die Daten angezeigt werden, die während der letzten VI-Ausführung durch das Verbindungsstück geflossen sind. Klicken Sie auf die Schaltfläche **Verbindungswerte erhalten**, um die Daten an jedem Punkt der Ausführung zu speichern, so dass Sie sofort die neuesten Werte der Daten erhalten, die übermittelt werden, wenn Sie eine Sonde an die Verbindung anlegen. Sie müssen das VI mindestens einmal erfolgreich ausführen, bevor Sie mit Hilfe von Sonden Daten von den Verbindungen sammeln können. Das Verwenden von Sonden nach der VI-Ausführung erweist sich als sinnvoll, wenn das Blockdiagramm sehr unübersichtlich ist und Sie das VI auf Fehler untersuchen möchten.



Um Verbindungswerte für Sonden beizubehalten, klicken Sie in der Blockdiagramm-Symbolleiste auf die Schaltfläche **Verbindungswerte erhalten**.



Sie können jederzeit die Schaltfläche **Verbindungswerte nicht erhalten** anklicken, um das Speichern von Verbindungswerten bei der Ausführung zu unterbinden.



Hinweis Wenn Sie möchten, dass Ihr VI weniger Speicherplatz belegt oder schneller läuft, aktivieren Sie die Option **Verbindungswerte nicht speichern**.

Profil-VIs in mehreren Zielsystemen

Mit Hilfe des Fensters **Profil - Leistung und Speicher** können Sie gleichzeitig VI-Profile auf verschiedenen Systemen erstellen. Das Fenster **Profil - Leistung und Speicher** enthält eine neue Spalte mit dem Namen **Projektbibliothek**, in der die LabVIEW-Projektbibliothek (* .lvlib) und

Zielsysteme angezeigt werden. In dieser Spalte können Sie zwischen VIs unterscheiden, die den gleichen Namen haben, aber in verschiedenen Projektbibliotheken enthalten bzw. auf verschiedenen Systemen geladen werden. Außerdem gibt es in dem Fenster eine neue Schaltfläche, **Applikationsinstanzen auswählen**, mit der das Dialogfeld **Applikationsinstanzen auswählen** geöffnet wird. Hier können Sie die Systeme angeben, die unter **Profil - Leistung und Speicher** untersucht werden sollen. Neben der Schaltfläche **Applikationsinstanzen auswählen** befindet sich eine Legende mit den Systemen, auf die vom Fenster **Profil - Leistung und Speicher** aus zugegriffen werden kann, und der Farben, mit denen sie gekennzeichnet sind. Das Fenster **Profil - Leistung und Speicher** wird nicht mehr wie bisher über **Werkzeuge»Fortgeschritten»Profil-VIs** geöffnet, sondern über **Werkzeuge»Profil»Leistung und Speicher**.



Hinweis Ein LabVIEW-Projekt (*.lvproj) ist eine Zusammenstellung von VIs, Build-Spezifikationen usw. Alle Dateien zu einem Projekt werden im Fenster **Projekt-Explorer** angezeigt. Es kann immer nur ein Projektprofil erstellt werden. Ist das Fenster **Profil - Leistung und Speicher** bereits geöffnet, ist die Option **Werkzeuge»Profil»Leistung und Speicher** deaktiviert.

64-Bit-Integer-Datentyp

Große Integer werden im 64-Bit-Integer-Datentyp (mit oder ohne Vorzeichen) verarbeitet, gespeichert und angezeigt.

Der 64-Bit-Integer-Datentyp wird von allen Funktionen mit polymorphen numerischen Eingängen unterstützt. Für die Rotation, das Schieben und die Multiplikation mit den Funktionen “Rotieren”, “Logisches Schieben” und “Potenz von 2 multiplizieren” werden 64-Bit-Eingänge in 32-Bit-Integer umgewandelt. Wenn die Funktionen einen 64-Bit-Integer ausgeben sollen, müssen Sie einen 64-Bit-Integer mit dem Eingang **String** der Funktionen “Dezimal-String nach Zahl”, “Hexadezimal-String nach Zahl”, “Oktal-String nach Zahl” und “Bruch-/Exponential-String nach Zahl” verbinden.

Mit Hilfe der Funktionen “Nach Quad-Integer” bzw. “Nach vorzeichenlosen Quad-Integer” lassen sich Zahlen in 64-Bit-Integer (mit Vorzeichen und vorzeichenlos) umwandeln.

NI Spy

NI Spy ist ein Programm zur Überwachung des Datenaustauschs mit Messinstrumenten, das zusammen mit einer Applikation ausgeführt wird. Mit NI Spy können während der Ausführung von VIs die Ein- und Ausgabeaufrufe der Messgeräte und die Ergebnisse des Aufrufs aufgezeichnet werden. Das Protokollieren von Aufrufen und deren Ergebnissen kann bei der Fehlerdiagnose beim Datenaustausch mit Messinstrumenten von Vorteil sein.

(Windows) Zum Starten von NI Spy wählen Sie **Start»National Instruments»NI Spy**.

(Mac OS) Zum Starten von NI Spy wählen Sie **Applications»National Instruments»NI Spy** und klicken Sie doppelt auf das Symbol von NI Spy.

(Linux) Geben Sie zum Starten von NI Spy `nispy` in die Befehlszeile ein.

Suchen und Ersetzen

In LabVIEW 7.x und Vorgängerversionen können Sie ein Blockdiagramm-Objekt ersetzen, indem Sie das entsprechende Objekt mit der rechten Maustaste anklicken und aus dem Kontextmenü **Ersetzen** auswählen. In LabVIEW 8.0 können Sie mehrere Objekte oder Textbereiche auf einmal ersetzen. Zum Durchsuchen von VIs nach anderen VIs, Objekten und Text gibt es das Dialogfeld **Suchen**. Im Fenster **Suchergebnisse** lassen sich die Suchtreffer ersetzen.

Individuelle Tastenkombinationen

Auf der Seite **Menü-Verknüpfungen** im Dialogfeld **Optionen** können Sie Tastenkombinationen für VI-Menüpunkte festlegen. Die Funktionstasten <F1> bis <F24> können ohne die <Strg>-Taste als Zusatztaste verwendet werden.

Neuerungen im Dialogfeld “Drucken”

Im Dialogfeld **Drucken** gab es folgende Verbesserungen:

- Auf der Seite **VI(s) auswählen** können Sie festlegen, ob das aktuelle oder mehrere VIs ausgedruckt werden sollen.
- Mit den Optionen **Vollständiges Frontpanel**, **Sichtbarer Bereich des Frontpanels** und **VI-Dokumentation** auf der Seite **Inhalt drucken** können Sie den Druckbereich festlegen. Die Optionen **Panel verwenden**, **Als SubVI verwenden**, **Vollständige Dokumentation** und **Angepasst** wurden von der Seite **Inhalt drucken** auf die Seite **VI-Dokumentation** verschoben. Diese können Sie durch Auswahl der Option **VI-Dokumentation** öffnen.
- Die Seite **Alle Bedienelemente** wurde in **VI-Dokumentation** umbenannt. Durch die Option **VI-Dokumentationsstil** können die Komponenten auf der Seite **VI-Dokumentation** festgelegt werden. Die Option **Symbol und Beschreibung** und deren Komponenten wurden durch die Optionen **Symbol und Anschlussfeld** und **Beschreibung** ersetzt. Mit den Komponenten **Beschriftung**, **Untertitel** und **Untertitel [Beschriftung]** der Option **Elemente** können Sie Beschriftungen und/oder Untertitel für Bedien- und Anzeigeelemente gedruckt werden.

- In diesem Dialogfeld werden immer die VIs der aktuellen Kopie aller geöffneten Anwendungen gedruckt. Zum Drucken von VIs in mehreren Kopien einer Anwendung ist der Druckvorgang in jeder geöffneten Applikationsinstanz zu wiederholen.

Bearbeiten von VI-Symbolen

Die VI-Vorlage "lv_icon" im Verzeichnis labview\resource\plugins dient zum Erstellen eines VIs, mit dem sich VI-Symbole außerhalb des Dialogfelds **Symboleditor** bearbeiten lassen.

Neuerungen und Änderungen an der Dokumentation

Nachfolgend sind alle Verbesserungen und Änderungen an der Dokumentation beschrieben.

Neugestaltung der Dokumentation

Die *LabVIEW-Hilfe* enthält nun viele Teile der Handbücher und Application Notes aus LabVIEW 7.x und Vorgängerversionen. Daher wird LabVIEW nicht mit einer PDF-Bibliothek oder PDF-Versionen von Dokumenten wie *LabVIEW Measurements Manual*, *Using External Code in LabVIEW*, *LabVIEW Development Guidelines*, *LabVIEW Analysis Concepts* oder Application Notes geliefert.

In der folgenden Tabelle sehen Sie den Speicherort der einzelnen Dokumente in LabVIEW 7.x und LabVIEW 8.0:

Tabelle 5. Änderungen an der Dokumentation in LabVIEW 8.0

Dokumentation	LabVIEW 7.x	LabVIEW 8.0
<i>Erste Schritte mit LabVIEW</i>	Druck- und PDF-Version	Druck- und PDF-Version
<i>LabVIEW-Versionshinweise</i>	Druck- und PDF-Version	Druck- und PDF-Version
<i>Hinweise zum Upgrade von LabVIEW</i>	Druck- und PDF-Version	Druck- und PDF-Version
<i>LabVIEW-Schnellübersicht</i>	Druck- und PDF-Version	Druck- und PDF-Version
<i>LabVIEW-Benutzerhandbuch</i>	Druck- und PDF-Version	<i>LabVIEW-Hilfe: Grundlagen</i> (Teile als Printversion)
<i>LabVIEW Measurements Manual</i>	Druck- und PDF-Version	<i>LabVIEW-Hilfe: Messungen durchführen; Instrumente steuern</i>

Tabelle 5. Änderungen an der Dokumentation in LabVIEW 8.0 (Fortsetzung)

Dokumentation	LabVIEW 7.x	LabVIEW 8.0
<i>Readme zum Application Builder für LabVIEW</i>	Druck- und PDF-Version (Benutzerhandbuch zum LabVIEW Application Builder)	labview\readme
<i>LabVIEW Development Guidelines</i>	Druck- und PDF-Version	LabVIEW-Hilfe: Grundlagen» Entwicklungsrichtlinien
<i>LabVIEW Analysis Concepts</i>	PDF	LabVIEW-Hilfe: Grundlagen» Signalverarbeitung und -analyse
<i>Using External Code in LabVIEW</i>	PDF	LabVIEW-Hilfe: Grundlagen» Aufrufen textbasierter Programmcodes
<i>Integrating the Internet into Your Measurement System—DataSocket Technical Overview</i>	PDF	—
<i>LabVIEW und Hyperthreading</i>	PDF	LabVIEW-Hilfe: Grundlagen» LabVIEW und Hyperthreading
<i>Benutzerdefinierte Bedien- und Anzeigeelemente und Typdefinitionen in LabVIEW</i>	PDF	LabVIEW-Hilfe: Grundlagen» Erstellen des Frontpanels» Allgemeines» Erstellen benutzerdefinierter Bedien- und Anzeigeelemente und Typdefinitionen
<i>LabVIEW-Datenspeicherung</i>	PDF	LabVIEW-Hilfe: Wie LabVIEW Daten zwischenspeichert
<i>LabVIEW Leistungs- und Speichermanagement</i>	PDF	LabVIEW-Hilfe: Grundlagen» Leistungs- und Speichermanagement» Allgemeines» Verwenden des VI-Profilfensters zur Überwachung der VI-Leistung
<i>Polymorphe Einheiten in LabVIEW</i>	PDF	LabVIEW-Hilfe: Grundlagen» Erstellen von VIs und SubVIs» Allgemeines» Verwendung polymorpher Einheiten
<i>Portieren und Lokalisieren von LabVIEW-VIs</i>	PDF	LabVIEW-Hilfe: Grundlagen» Portieren und Lokalisieren von VIs

Tabelle 5. Änderungen an der Dokumentation in LabVIEW 8.0 (Fortsetzung)

Dokumentation	LabVIEW 7.x	LabVIEW 8.0
<i>AppleEvents und die PPC Toolbox zur Kommunikation mit LabVIEW-Anwendungen auf Macintosh-Rechnern</i>	PDF	—
<i>Verwendung von LabVIEW zum Erstellen von Multithread-VIs für maximale Leistung und Zuverlässigkeit</i>	PDF	<i>LabVIEW-Hilfe: Grundlagen»Definitionen: Multitasking, Multithreading und Multiprocessing</i>
<i>Verwenden von LabVIEW mit TCP/IP und UDP</i>	PDF	<i>LabVIEW-Hilfe: Verwenden von LabVIEW mit TCP/IP und UDP</i>
<i>Verwenden von LabVIEW mit drahtlosen Geräten</i>	PDF	<i>LabVIEW-Hilfe: Verwenden von LabVIEW mit drahtlosen Geräten</i>
<i>Using the Timed Loop to Write Multirate Applications in LabVIEW</i>	PDF	<i>LabVIEW-Hilfe: Grundlagen»Schleifen und Strukturen»Zeitgesteuerte Strukturen</i>

(Windows) Führen Sie folgende Schritte aus, um ein komplettes Buch aus der Registerkarte **Inhalt** der *LabVIEW-Hilfe* zu drucken:

1. Klicken Sie das Buch mit der rechten Maustaste an.
2. Wählen Sie aus dem Kontextmenü die Option **Drucken** aus, um zum Dialogfeld **Themen drucken** zu gelangen.
3. Klicken Sie auf **Ausgewähltes Thema und alle Unterthemen drucken**.



Hinweis Zum Drucken eines einzelnen Themas, das Sie auf der Registerkarte **Inhalt** ausgewählt haben, wählen Sie **Ausgewähltes Thema drucken**.

4. Klicken Sie auf **OK**.

In dieser Hilfedatei können Links zu PDF-Dateien enthalten sein. Zum Ausdrucken einer PDF-Datei klicken Sie auf die Schaltfläche “Drucken” in der Symbolleiste des Adobe Acrobat Viewers.

Im [Adobe Reader mit Suchfunktion ab 6.x](#) können Sie die PDF-Versionen des Buchs *LabVIEW-Grundlagen*, der *LabVIEW-Versionshinweise*, der *Hinweise zum Upgrade von LabVIEW*, des Handbuchs *Erste Schritte mit LabVIEW* und der *LabVIEW Application Builder*

Readme durchsuchen. Gehen Sie zum Durchsuchen aller bei der Lieferung von LabVIEW enthaltenen PDF-Dokumente wie folgt vor:

1. Wählen Sie **Bearbeiten»Suchen**, um zum Fenster **PDF durchsuchen** zu gelangen.
2. Geben Sie in das Feld **Nach welchem Wort/welcher Wortgruppe suchen Sie?** den gewünschten Suchbegriff ein.
3. Klicken Sie auf die Schaltfläche **Alle PDF-Dokumente in** und wählen Sie das Verzeichnis `labview\manuals` aus.
4. Klicken Sie auf die Schaltfläche **Suchen**.

Weitere Informationen zur Volltextsuche in allen PDF-Dokumenten eines Ordners finden Sie in der Hilfe zum Adobe Reader.

Readme-Dateien (nur Windows)

Das Verzeichnis *readme* in der LabVIEW-Dateistruktur enthält die Readme-Dateien für LabVIEW und alle anderen installierten Module und Toolkits. Die Readme-Dateien befinden sich unter **Start»Alle Programme»National Instruments»LabVIEW»Readme**.

Schaltfläche “Anzeigen”

Für die Themensuche in der *LabVIEW-Hilfe* gibt es drei Registerkarten: **Index**, **Suchen** und **Favoriten**. Mit der Schaltfläche **Anzeigen** wird das Thema im Inhaltsverzeichnis gesucht.



Hinweis Das Inhaltsverzeichnis der LabVIEW-Hilfe enthält keine Beschreibungen zu Dialogfeldern.

Zugriffsmöglichkeiten

Sie können zum Öffnen eines Hilfethemas mit einem Link zu einem Beispiel für ein VI oder eine Funktion dieses VI oder die Funktion auch mit der rechten Maustaste auf dem Blockdiagramm oder in einer fixierten Palette anklicken und aus dem Kontextmenü **Beispiele** auswählen. Mit der Reißzwecke in der linken oberen Ecke einer Palette lässt sich diese auf dem Bildschirm fixieren.

Kontexthilfe

Im Fenster **Kontexthilfe** wurden der Link **Klicken Sie hier, um mehr Hilfe zu erhalten** und die Schaltfläche **Weitere Hilfe** in **Ausführliche Hilfe** umbenannt.

Neuerungen an der Beispielsuchmaschine

In der Liste **Vorraussetzungen** auf den Registerkarten **Suchen** und **Index** der Suchmaschine für Beispiele wird die Hard- und Software angezeigt, die für ein Beispiel benötigt wird. Zum Anzeigen der Liste **Anforderungen** in der Suchmaschine für Beispiele aktivieren Sie die Option **Liste mit Anforderungen anzeigen** auf der Registerkarte **Allgemein** des Dialogfelds **Einstellungen zur Beispielsuchmaschine**.

Mit Hilfe der Suchmaschine für Beispiele können Sie Kommentare und Bewertungen für Web-Beispiele in der **NI Developer Zone** anzeigen lassen und einsenden. Die Suchmaschine enthält auch Links zu ähnlichen Dokumenten für Web-Beispiele. Zur Anzeige der Liste **Ressourcen** aktivieren Sie die Option **Ressourcen-Liste anzeigen** auf der Registerkarte **Web** des Dialogfelds **Einstellungen zur Beispielsuchmaschine**.

Auf der Registerkarte **Suchen** der Suchmaschine wird in der NI Developer Zone nach Beispielen zu einer bestimmten Problemstellung gesucht. **(Windows und Linux)** Zum Suchen nach Web-Beispielen wählen Sie die Option **Thema** im Abschnitt **Suchkriterium** und aktivieren die Option **Beispiele auf ni.com einbeziehen**. Zum Festlegen der maximalen Anzahl der angezeigten Web-Beispiele verwenden Sie das Feld **Maximale Anzahl der Suchergebnisse im Web** auf der Registerkarte **Web** im Dialogfeld **Einstellungen zur Beispielsuchmaschine**. Sie können auch das Feld **Zeitlimit für Web-Suche (Sekunden)** auf der Registerkarte **Web** im Dialogfeld **Einstellungen zur Beispielsuchmaschine** verwenden, um festzulegen, wie lange die NI-Suchmaschine für Beispiele im Web suchen soll.

Bei der themenbezogenen Suche enthält der Ordner **Nicht installierte Toolkits und Module** in der Liste **Zum Öffnen eines Beispiels klicken Sie es doppelt an** Beispiele zu nicht installierten Toolkits und Modulen. Wenn Sie aus diesem Ordner ein Beispiel auswählen, erscheint in der Liste **Voraussetzungen** ein Link für weitere Informationen zum betreffenden Toolkit oder Modul.

Mit der Registerkarte **Index** ist eine Suche nach vorinstallierten Beispielen per Stichwort oder Beschreibung möglich. Zur Suche nach Schlüsselwörtern wählen Sie aus dem Pulldown-Menü neben der Schaltfläche **Index** die Option **Schlüsselwort** aus. Wählen Sie die Option **Beschreibung** aus dem Pulldown-Menü neben der Schaltfläche **Index**, um nach Beschreibungen zu suchen. Zur Beispielsuche nach Beschreibungen müssen Sie jedoch die Option **Beispiele auf ni.com einbeziehen** deaktivieren. Wenn Sie die Beispiele nach Beschreibungen durchsucht haben, wird der Text, nach dem Sie gesucht haben, im Feld **Information** hervorgehoben.

Mit Hilfe der NI-Suchmaschine für Beispiele können Sie nach Beispielen in einzelnen VIs oder Projekten suchen. Wenn Sie ein Beispiel in einem Projekt öffnen, wird die entsprechende LabVIEW-Projektdatei

(* .lvproj) im Fenster **Projekt-Explorer** geöffnet. Klicken Sie zum Öffnen des Beispiels das VI unter **Mein Computer** im Projekt-Explorer doppelt an. Bei den üblichen VI-Beispielen hat sich nichts geändert.

In LabVIEW 7.x und Vorgängerversionen kann die Suchmaschine für Beispiele nur installierte DAQ-Karten finden. In LabVIEW 8.0 findet die NI-Suchmaschine für Beispiele die meisten installierten Geräte. Zum Hinzufügen nicht installierter Geräte klicken Sie die Schaltfläche **Einstellungen**, um zum Dialogfeld **Einstellungen zur Beispielsuchmaschine** zu gelangen.

Andere Änderungen und Neuerungen in LabVIEW 8.0

Nachfolgend alle weiteren Änderungen an LabVIEW 8.0:

Änderungen an VIs und Funktionen

Hinsichtlich der vorhandenen VIs und Funktionen gibt es in LabVIEW 8.0 folgende Änderungen:

- Der Ausgang **VISA-Ressourcen-Name** der VIs “VISA: Seriellen Port konfigurieren” und “VISA: Serielle Übertragung unterbrechen” wurde in **VISA-Ressourcen-Name (Ausgang)** umbenannt. Der Ausgang **VISA USB Intr Ereignis(Kopie)** des VIs “VISA: USB-Interruptdaten abfragen” wurde in **VISA-USB: Intr Ereignis (Ausgang)** umbenannt. Der Ausgang **VISA-Ressourcen-Name** aller anderen VISA-VIs und -Funktionen wurde in **VISA-Ressourcen-Name (Ausgang)** umbenannt.
- Der Eingang **Protokoll** der Funktion “VISA: Trigger freigeben” hat zwei neue Werte: 6 (PXI: Reservieren) und 7 (PXI: Freigeben).
- Das VI “VISA: Asynchron sperren” hat zwei neue Eingänge, und zwar **Sperrtyp** und **angeforderter Schlüssel**, sowie den neuen Ausgang **Zugangsschlüssel**. Mit diesen Ein- und Ausgängen können Sie eine exklusive oder gemeinsame Sperre für die aktuelle Session anfordern.
- Der Eingang **Dauer** des VIs “VISA: Serielle Übertragung unterbrechen” gibt die Dauer der Unterbrechung in Millisekunden an.
- Der Eingang **Ereignis-Ressourcenklasse** der Funktion “VISA: Auf Ereignis warten” wurde in **Ereignisklasse** umbenannt und kann jetzt offen gelassen werden.
- Die Funktion “Signalverlaufskomponente lesen” ruft standardmäßig das **Y**-Datenarray ab.
- Wenn Sie die Funktion “Mehrfacharithmetik” aus der Palette **Numerisch** im Blockdiagramm ablegen, lautet die Voreinstellung “Addieren”. Beim Einfügen der gleichen Funktion von der Palette **Boolesch** lautet die Voreinstellung “ODER”.

- Die Funktionen “In Variant”, “Variant nach Daten”, “Variant nach String” und “Variant-Attribut setzen” arbeiten nun schneller. Die meisten VIs, die diese Funktionen aufrufen, werden jetzt ebenfalls schneller ausgeführt. Die Leistung wurde jedoch besonders in VIs mit großen Datensätzen verbessert.
- Der Parameter **String** der Funktion “String suchen und ersetzen” wurde in **Eingangsstring** umbenannt. Die Funktion “String suchen und ersetzen” enthält nun fünf neue Parameter: Unter **Schreibweise ignorieren?** wird angegeben, ob die Groß- und Kleinschreibung bei der Suche eine Rolle spielen soll. Unter **Mehrzeilig?** wird angegeben, ob die Zeichen ^ und \$ jeweils am Anfang und Ende einer Zeile oder des ganzen Strings gefunden werden sollen. Unter **Anzahl der Ersetzungen** erfahren Sie, wie oft LabVIEW den **Suchstring** ersetzt hat. Die Funktion hat nun außerdem einen **Fehler (Eingang)** und **Fehler (Ausgang)**.
- Der Eingang **Transparenz-Schwellwert** des VIs “PNG-Datei lesen” fügt Alpha-Werte eines 32-Bit-PNG-Bildes in **Maskieren** der resultierenden **Grafikdaten**. Alle Alpha-Werte, die größer oder gleich dem **Transparenz-Schwellwert** sind, werden von LabVIEW als nicht transparent behandelt. Alle anderen Alpha-Werte werden von LabVIEW als vollständig transparent behandelt. Der **Transparenz-Schwellwert** muss zwischen 0 und 225 liegen.
- Am Eingang **Vergleichsmodus** des VIs “Nach Digitalmustern suchen” wird festgelegt, wie Werte von X für die Suche bearbeitet werden sollen. Die Eingänge **Modus** und **Startindex/Startzeit** der Instanz “DWDT: Nach Digitalmustern suchen” dieses VIs wurden in **Startwertformat** und **Start** umbenannt. Der **Index-Modus** heißt nun **Abtastrate**.
- Die Eingänge **Modus** und **Startindex/Startzeit** der Signalverlaufsinstanzen des VIs “y-Wert lesen” wurden in **Format der Y-Position** und **Y-Position** umbenannt. Der Ausgang **Tatsächlicher Index/Zeit-Wert** dieser Instanzen wurde in **tatsächliche Y-Position** umbenannt.
- Der Ausgang **Signalverlaufsdatenwert** der Instanz “DWDT: Y-Wert lesen” des VIs “y-Wert lesen” gibt die Werte der Signalverlaufsdaten aus.
- Der Eingang **digitaler Datenwert** der Instanz “DTbl: Y-Wert lesen” des VIs “y-Wert lesen” gibt die Werte der digitalen Daten aus. Der Ausgang **Signalauschnitt** dieser Instanz wurde in **Ausschnitt digitaler Daten** umbenannt. Der Ausgang **Digital (Ausgang)** dieser Instanz wurde in **Digitaldaten (Ausgang)** umbenannt.
- Der Eingang **Modus** der Instanzen “WDT: Signalverlaufsabschnitt lesen” und “DWDT: Signalverlaufsabschnitt” des VIs “Signalverlaufsabschnitt lesen” wurde in **Format von Start/Dauer** umbenannt. Der

Index-Modus heißt nun **Abtastrate**. Die Instanz “WDT: Signalverlaufsabschnitt lesen” hat auch ein neues **Format von Start/Dauer**, und zwar **Absolute Zeit**.

- Der Eingang **Digital** der Instanz “DTbl: Digital: Signalausschnitt” des VIs “Signalverlaufsabschnitt” wurde in **Digitaldaten** umbenannt. Der Ausgang **Digital: Signalausschnitt** dieser Instanz wurde in **Ausschnitt digitaler Daten** umbenannt.
- Die VIs “y-Wert lesen”, “Signalverlauf in XY-Paare konvertieren”, “Signalverlaufsarray indizieren”, “Signalverlaufs-Zeitstempel ausrichten”, “Signalverläufe anhängen”, “Signalverlaufs-dt kopieren” und “Delta-t skalieren” haben jetzt Instanzen für vorzeichenbehaftete 64-Bit-Integer. Das VI “Signalverlauf in XY-Paare konvertieren” hat auch eine Instanz für vorzeichenlose 64-Bit-Integer.
- Das VI “Boolesches Array nach Digitaldaten” hat zwei neue Eingänge. Am Eingang **Daten komprimieren** wird festgelegt, ob die digitalen Ausgabedaten komprimiert werden sollen, und unter **Abtastrate** wird die Frequenz des ausgegebenen digitalen Signalverlaufs in Samples pro Sekunde angegeben.
- Der Eingang **Auf Fehler reagieren** des VIs “Digital: Invertieren” wurde in **Reaktion auf Invertierungsfehler** umbenannt. Das VI “Digital: Invertieren” verfügt nun über zwei neue Eingänge. Unter **Signalindex** wird das Signal festgelegt, an dem mit dem Invertieren der Daten begonnen werden soll, und unter **Anzahl der Signale** wird die Anzahl der zu invertierenden Signale angegeben, beginnend mit dem Signal am **Signalindex**.
- Am Eingang **Systemwarnung?** des VIs “Signalton” wird festgelegt, ob die Standardwarnung verwendet werden soll. Wenn ja, werden **Frequenz (Hz)** und **Dauer (ms)** ignoriert. Den Eingang **Intensität** und den Ausgang **Fehler** aus LabVIEW 7.x und Vorgängerversionen gibt es nicht mehr.
- Wenn der Eingang **Eingabeaufforderung zum Ersetzen?** des VIs “Report in Datei speichern” TRUE und der **Report-Dateipfad** ein Pfad zu einer bereits vorhandenen Datei ist, öffnet sich ein Dialogfeld und Sie werden gefragt, ob die bereits vorhandene Datei ersetzt werden soll. Andernfalls wird die bereits vorhandene Datei ohne Warnung überschrieben.
- Die Voreinstellung **Maßeinheit für Spaltenbreite** für das VI “Tabelle an Report anfügen” lautet **Standard**. Wenn Sie mit Hilfe dieses VIs eine Tabelle an einen HTML-Report anfügen, ignoriert LabVIEW den Eingang **Maßeinheit für Spaltenbreite**. Stattdessen wird die **Spaltenbreite** mit 100 multipliziert, um die Spaltenbreite in Pixeln zu ermitteln.

- Am Eingang **Konfigurationsdaten der Express-VIs hinzufügen** des VIs “Liste der SubVIs an Report anfügen” wird angegeben, ob im Report Konfigurationsinformationen zu den Express-VIs des Blockdiagramms enthalten sein sollen.
- Zur Palette **Strings** wurde eine Leerzeichen-Konstante hinzugefügt. Die Leerzeichen-Konstante liefert einen String von einem Leerzeichen für das Blockdiagramm.
- Die Funktion “In String konvertieren” hat zwei neue Eingänge: Unter **Array- oder Stringgröße voranstellen?** wird angezeigt, ob die Größe des Arrays oder Strings dem **Datenstring** vorangestellt werden soll, sofern es sich bei **Beliebiges Objekt** um ein Array oder einen String handelt. An **Byte-Reihenfolge** erhalten Sie die Byte-Folge des ausgegebenen Strings. Mit den Voreinstellungen dieser neuen Eingänge verhält sich die Funktion wie gehabt. Die Funktion hat nun außerdem einen **Fehler (Eingang)** und **Fehler (Ausgang)**.
- Bei der Funktion “In String konvertieren” gibt es einen neuen Kontextmenüpunkt, **7.x-Daten konvertieren**. Mit Hilfe dieser Option wird der Ausgang **Typen-String (nur 7.x)** und das Symbol für diese Funktion (rotes 7.x) angezeigt. Der Menüpunkt **Typdefinitionen darstellen** ist nur zu sehen, wenn Sie den Eingang **Typen-String (nur 7.x)** mit einem Typen-String verbinden. Der Ausgang **Typen-String** wurde in **Typen-String (nur 7.x)** umbenannt. Der Ausgang wird nur angezeigt, wenn er mit Daten von früheren LabVIEW-Versionen verbunden ist oder Sie mit der rechten Maustaste auf die Funktion klicken und den Menüpunkt **7.x-Daten konvertieren** auswählen.
- Die Funktion “String in Daten konvertieren” hat zwei neue Eingänge. Mit **Array- oder String-Größe voranstellen?** wird festgelegt, ob die Größe des Arrays oder Strings am Anfang der Daten stehen soll. Unter **Byte-Folge** wird die Byte-Folge des Eingangsstrings angezeigt. Mit den Voreinstellungen dieser neuen Eingänge verhält sich die Funktion wie gehabt. Die Funktion hat auch einen neuen Ausgang, **Rest des binären Strings**, der alle übrig gebliebenen Bytes enthält, die die Funktion nicht umgewandelt hat. Der Ausgang **Fehler** wurde in **Fehler (Eingang)** und **Fehler (Ausgang)** aufgeteilt.
- Bei der Funktion “Datum/Zeit nach Sekunden” ist der Eingang **Weltzeit** hinzugekommen. Dieser gibt an, ob **Datum/Zeit** in Weltzeit oder in der eingestellten Zeitzone des Computers angegeben wird. In LabVIEW 7.x wurde der Parameter **Sommerzeit** von der Funktion “Sekunden nach Datum/Zeit” ignoriert, da die Funktion Berechnungen auf Grundlage der Weltzeit durchführt. Wenn **Weltzeit** FALSE ist (Voreinstellung), wandelt LabVIEW 8.0 die Zeit basierend auf **Sommerzeit** um. Wenn Sie ein VI öffnen, das in LabVIEW 7.x oder Vorgängerversionen gespeichert wurde, wird **Sommerzeit** auf -1 eingestellt, um die Umwandlung zu vermeiden.

- Bei den Funktionen “Datum/Zeit nach Sekunden” und “Sekunden nach Datum/Zeit” gibt es im Cluster **Datum/Zeit** ein neues Element namens **Sekundenbruchteile**, das angibt, wie viele Sekundenbruchteile seit Beginn der Sekunde verstrichen sind.
- Der Eingang **in Weltzeit** der Funktion “Sekunden nach Datum/Zeit” legt fest, ob **Datum/Zeit** in Weltzeit oder in der eingestellten Zeitzone des Computers angegeben werden soll.
- Bei der Funktion “Datum/Zeit formatieren” ist der Eingang **Weltzeit** hinzugekommen. Dieser gibt an, ob die Zeit in Weltzeit oder in der eingestellten Zeitzone des Computers angegeben werden soll.
- Die Funktionen “In String formatieren” und “In String suchen” arbeiten jetzt auch mit Daten des Typs “Zeitstempel”. Im Dialogfeld **Format-String bearbeiten** gibt es die Konvertierungsoptionen **Format für relative Zeit** und **Zeitstempel** und im Dialogfeld **Such-String bearbeiten** gibt es jetzt die Optionen **Relative Zeit suchen** und **Zeitstempel suchen**.
- Mit Hilfe des Syntaxelements für Formatbezeichner \$ lässt sich mit der Funktion “In String formatieren” die Reihenfolge der Variablen im Format-String festlegen.
- Der Eingang **Erzeugen, wenn nicht gefunden?** der Instanzen “String”, “DBL”, “SGL”, “I32”, “I16”, “U8” und “Zeitstempel” des VIs “Eigenschaft setzen” legt fest, ob eine benutzerspezifische Eigenschaft erstellt werden soll, wenn es die unter **Eigenschaftsname** angegebene Eigenschaft nicht gibt.

Neue Beispiel-VIs

Im Ordner **Neue Beispiele für LabVIEW 8.0** auf der Registerkarte **Suchen** der NI-Beispielsuchmaschine finden Sie neue Beispiel-VIs und Beschreibungen dazu. Die Beispiele können direkt gestartet werden.

Änderungen an externen Code-Funktionen

Die folgenden neuen Funktionen können von einer DLL oder einem CIN aus aufgerufen werden:

- `CToLStr` wandelt einen C-String in einen LabVIEW-String um.
- `LToCStr` wandelt einen LabVIEW-String in einen C-String um.
- `DSSetHandleFromPtrNULLMeansEmpty` weist einem Zeiger ein Handle zu und kopiert die Daten aus dem Zeiger in das Handle. Diese Funktion ist vergleichbar mit der Funktion `DSSetHandleFromPtr`. Wenn der Zeiger jedoch keine Daten hat, setzt die Funktion `DSSetHandleFromPtrNULLMeansEmpty` das Handle auf Null, anstatt dem Handle Null Bytes zuzuweisen.

- `NIGetOneErrorCode` wandelt einen numerischen Fehlercode in die dazugehörige Fehlermeldung um.

Die CIN-Funktionen `AZHPurge`, `AZHNoPurge`, `AZHLock` und `AZHUnlock` aus LabVIEW 7.x und Vorgängerversionen werden nicht mehr unterstützt.

Eigenschaften und Methoden mit neuem Namen

In LabVIEW 8.0 gibt es neue VI-Server-Klassen, -Eigenschaften, -Methoden und -Ereignisse. Die nachfolgend aufgeführten Eigenschaften und Methoden wurden ebenfalls in LabVIEW 8.0 umbenannt.

Eigenschaften

Die folgenden Eigenschaften wurden umbenannt:

Tabelle 6. Neue Eigenschaftsnamen in LabVIEW 8.0

Klasse	Name in LabVIEW 7.1	Name in LabVIEW 8.0
Applikation	Applikation:Verbundener Real-Time-Host	Applikation:Benutzeroberfläche verfügbar
Binärtabelle	Zellenhintergrundfarbe	Aktive Zelle:Zellenhintergrundfarbe
Binärtabelle	Zellenvordergrundfarbe	Aktive Zelle:Vordergrundfarbe der Zelle
Binärtabelle	Links oben sichtbare Zelle	Zelle links oben
Listefeld	Bildlaufleiste sichtbar	Sichtbare Objekte:Vertikale Bildlaufleiste sichtbar
Listefeld	Symbole sichtbar	Sichtbare Objekte:Symbole sichtbar
MehrspaltigesListefeld	Zellenhintergrundfarbe	Aktive Zelle:Zellenhintergrundfarbe
MehrspaltigesListefeld	Zellengröße	Aktive Zelle:Zellengröße
MehrspaltigesListefeld	Spaltentitel sichtbar	Sichtbare Objekte:Spaltentitel sichtbar
MehrspaltigesListefeld	Horizontale Bildlaufleiste sichtbar	Sichtbare Objekte:Horizontale Bildlaufleiste sichtbar
MehrspaltigesListefeld	Symbole sichtbar	Sichtbare Objekte:Symbole sichtbar
MehrspaltigesListefeld	Vertikale Bildlaufleiste sichtbar	Sichtbare Objekte:Vertikale Bildlaufleiste sichtbar
Skalierung	Farbe der Feinteilung	Farben der Unterteilung:Farbe der Feinteilung
String	Bildlaufleiste sichtbar	Vertikale Bildlaufleiste sichtbar

Tabelle 6. Neue Eigenschaftsnamen in LabVIEW 8.0 (Fortsetzung)

Klasse	Name in LabVIEW 7.1	Name in LabVIEW 8.0
Tabelle	Zellenhintergrundfarbe	Aktive Zelle:Zellenhintergrundfarbe
Tabelle	Zellengröße	Aktive Zelle:Zellengröße
Tabelle	Spaltentitel sichtbar	Sichtbare Objekte:Spaltentitel sichtbar
Tabelle	Horizontale Bildlaufleiste sichtbar	Sichtbare Objekte:Horizontale Bildlaufleiste sichtbar
Tabelle	Index sichtbar	Sichtbare Objekte:Index sichtbar
Tabelle	Zeilentitel sichtbar	Sichtbare Objekte:Zeilentitel sichtbar
Tabelle	Vertikale Bildlaufleiste sichtbar	Sichtbare Objekte:Vertikale Bildlaufleiste sichtbar
Baumstruktur	Verschieben möglich	Ziehen/Ablegen:Verschieben von Objekten möglich
Baumstruktur	Einfügen zwischen Objekten zulassen	Ziehen/Ablegen:Ablegen zwischen Objekten zulassen
Baumstruktur	Ziehen übergeordneter Objekte zulassen	Ziehen/Ablegen:Ziehen übergeordneter Objekte zulassen
Signalverlaufdiagramm	Autom. Größenanpassung der Legende	Legende:Automatische Größenanpassung
Signalverlaufgraph	Autom. Größenanpassung der Legende	Legende:Automatische Größenanpassung

Methoden

Die folgenden Methoden wurden umbenannt:

Tabelle 7. Neue Methodennamen in LabVIEW 8.0

Klasse	Name in LabVIEW 7.1	Name in LabVIEW 8.0
Bedienelement	Neuinit. auf Std.	Standardwert wiederherstellen
Listenfeld	Doppelklick - Zeile ermitteln	Doppelt angeklickte Zeile ermitteln
MehrspaltigesListenfeld	Doppelklick - Zeile ermitteln	Doppelt angeklickte Zeile ermitteln
Text	An bestimmte Stelle verschieben	Zur Standardposition verschieben

Tabelle 7. Neue Methodennamen in LabVIEW 8.0 (Fortsetzung)

Klasse	Name in LabVIEW 7.1	Name in LabVIEW 8.0
Baumstruktur	Benutzer-Objektsymbole: Symbol zurücksetzen	Benutzer-Objektsymbole:Auf LabVIEW-Symbol zurücksetzen
Baumstruktur	Benutzer-Objektsymbole: Symbole zurücksetzen	Benutzer-Objektsymbole:Alle auf LabVIEW-Symbol zurücksetzen
Baumstruktur	Benutzer-Objektsymbole: Symbol setzen	Benutzer-Objektsymbole:Auf benutzerdefiniertes Symbol setzen
Baumstruktur	Benutzer-Objektsymbole: Symbol-Array setzen	Benutzer-Objektsymbole:Auf benutzerdefiniertes Symbol-Array setzen
VI	FP schließen	Frontpanel:Schließen
VI	VI-Strings exportieren	VI-Strings:Exportieren
VI	Alle Bedienelement-Werte ermitteln	Bedienelementwert:Alle lesen [Serialisiert]
VI	Alle Bedienelement-Werte ermitteln [Variant]	Bedienelementwert:Alle abfragen [Variant]
VI	Bedienelementwert ermitteln	Bedienelementwert:Lesen [Serialisiert]
VI	Bedienelement-Wert ermitteln [Variant]	Bedienelementwert:Abfragen [Variant]
VI	Skalierte Diagramm-Grafik lesen	Blockdiagramm:Skaliertes Bild lesen
VI	Sperrstatus ermitteln	Sperrstatus:Lesen
VI	Panel-Grafik abfragen	Frontpanel:Panel-Grafik abfragen
VI	Skalierte Panel-Grafik lesen	Frontpanel:Skalierte Panel-Grafik lesen
VI	VI-Symbol als Grafikdaten lesen	VI-Symbol:Als Bilddaten lesen
VI	VI-Strings importieren	VI-Strings:Importieren
VI	Netzwerkpanel-Steuerung sperrern	Netzwerkpanel:Für Steuerung sperrern
VI	Aktuelle Werte als Standard	Standardwerte:Aktuelle Werte als Standard
VI	FP öffnen	Frontpanel:Öffnen

Tabelle 7. Neue Methodennamen in LabVIEW 8.0 (Fortsetzung)

Klasse	Name in LabVIEW 7.1	Name in LabVIEW 8.0
VI	Standardwerte wiederherstellen	Standardwerte:Alle Standardwerte wiederherstellen
VI	VI-Symbol in Datei speichern	VI-Symbol:In Datei speichern
VI	Bedienelement-Wert setzen	Bedienelementwert:Setzen [Serialisiert]
VI	Bedienelement-Wert setzen [Variant]	Bedienelementwert:Festlegen [Variant]
VI	Sperrstatus setzen	Sperrstatus:Setzen
VI	VI-Symbol aus Datei lesen	VI-Symbol:Aus Datei lesen
VI	VI-Symbol aus Grafikdaten	VI-Symbol:Aus Bilddaten erzeugen
VI	Netzwerkpanel freigeben	Netzwerkpanel:Steuerung freigeben

Verbesserungen an den Fehlercodes

Die Fehlercodes –8999 bis –8000 stehen für eigene Fehlermeldungen zur Verfügung.

Im Hinblick auf Fehlercodes gibt es bei LabVIEW 8.0 folgende Änderungen:

Tabelle 8. Änderungen an den Fehlercodes in LabVIEW 8.0

Fehlercode in LabVIEW 7.1	Fehlercode in LabVIEW 8.0
20003	20012
20101	20111
20102	20112
20103	20113
20104	20114

Verschiedenes

Nachfolgend eine lose Auflistung sonstiger Änderungen an LabVIEW 8.0:

- Das Dialogfeld **Registrierungsinformationen**, welches bisher bei jedem Start einer nicht registrierten LabVIEW-Version angezeigt wurde, erscheint in LabVIEW 8.0 nicht mehr. LabVIEW 8.0 erfasst Informationen wie Ihren Namen und den Namen der Firma während der Installation.

- **(Windows)** In LabVIEW 7.x und Vorgängerversionen können mit der Tastenkombination <Strg-Tab> alle geöffneten LabVIEW-Fenster in der Reihenfolge, wie sie geöffnet wurden, angeschaut werden. Bei LabVIEW 8.0 können Sie sich die LabVIEW-Fenster mit der Tastenkombination <Strg-Tab> in der Reihenfolge ansehen, in der sie auf dem Bildschirm angezeigt werden. Diese Reihenfolge entspricht der Reihenfolge beim Drücken der Tastenkombination <Alt-Tabulator>. **(Linux)** Die Reihenfolge der Fenster hängt vom verwendeten Fenster-Verwaltungsprogramm ab.
- Wenn Sie ein Objekt in einem Baumstruktur-Element unter ein untergeordnetes Objekt ziehen, wird es unter dem untergeordneten Objekt auf der gleichen Hierarchieebene abgelegt.
- Mit Hilfe der Sonde für VI-Refnums (Panel-Grafik) wird der VI-Name, VI-Pfad, das Frontpanel und der Hexadezimalwert der Refnum angezeigt. Die Sonde lässt sich so konfigurieren, dass sie einen Haltepunkt setzt, wenn der Wert eine ungültige Refnum ist.
- Die Eigenschaft “Deaktivierte Elemente” für ein “Listenfeld” und “mehrspalziges Listenfeld” gibt eine Fehlermeldung aus, wenn sich die Eigenschaft auf ein Objekt bezieht, das sich nicht im Listenfeld befindet.
- **(Mac OS)** In LabVIEW kann mit dem Mausrad gescrollt werden. Um durch die Unterdiagramme einer Case-, Ereignis- oder gestapelten Sequenzstruktur zu scrollen, bewegen Sie den Cursor auf die Selektorkennung und drücken Sie während der Bewegung des Mausrads die <Command>-Taste.
- **(Mac OS)** LabVIEW reagiert nur bei Popup-Fenstern auf den <Control>-Klick. Die Kombination <Command>-Klick gibt es nicht mehr.
- Wenn Sie sich im Fenster **Bedienelement-Editor** befinden und das Bedienelement im Bearbeitungsmodus ist, können Sie ein Bild für einen Teil eines benutzerdefinierten Bedienelements aus einer Datei importieren, indem Sie aus dem Kontextmenü die Option **Aus Datei importieren** auswählen. Außerdem können Sie ein Gestaltungselement durch ein Bild aus dem Zwischenspeicher ersetzen, indem Sie das Gestaltungselement und anschließend **Bearbeiten»Einfügen** auswählen. Um das Gestaltungselement durch ein Bild aus einer Datei zu ersetzen, klicken Sie es mit der rechten Maustaste an und wählen Sie **Aus Datei importieren**.
- Es ist keine Angabe eines benutzerdefinierten **Menüverzeichnis** auf der Seite **Pfade** des Dialogfelds **Optionen** mehr möglich.
- Bei einem benutzerdefinierten **VI-Suchpfad** auf der Seite **Pfade** können Sie den symbolischen Pfad <osdatadir> verwenden.

- Wenn Sie einen Zeitstempel in ein Variant umwandeln, dann zeigt das Variant-Anzeigeelement den aktuellen Wert des Zeitstempels an, mit dem es verbunden ist.
- Der “CVI Funktionspanel-Konverter” und das Dialogfeld **VXIPNP-Treiber aktualisieren** sind in LabVIEW 8.0 nicht mehr vorhanden. Um diese Funktionen zu erhalten, laden Sie sich von ni.com/idnet das Gerätetreiber-Tool “LabVIEW Interface Generator for LabWindows/CVI” herunter.
- Die Option **Texteingabe mit Eingabe-Taste beenden** befindet sich auf der Seite **Umgebung** des Dialogfelds **Optionen**.
- Das Dialogfeld **Änderungen erklären** enthält ein Textfeld **Pfad**, in dem der Pfad des ausgewählten VIs auf dem Speichermedium aufgeführt wird. Mit Hilfe von **Pfad** können Sie VIs mit dem gleichen Namen unterscheiden.
- Die Ausführungsgeschwindigkeit von VIs mit deaktivierter Fehlersuche hat sich erhöht.
- Das Fenster **Verwaltung von VI-Bibliotheken** wurde in **LLB-Manager** umbenannt. Um ein Element in einer LLB der höchsten Ebene zuzuordnen, klicken Sie mit der rechten Maustaste auf die Liste **Dateien** und wählen Sie **Höchste Ebene?**. Änderungen, die Sie im LLB-Manager vorgenommen haben, können nicht rückgängig gemacht werden.
- Das Dialogfeld **VI-Bibliothek bearbeiten** wurde aus dem Menü **Werkzeuge** entfernt. Im Fenster **LLB-Manager** können Sie nun ein VI aus einer Bibliothek entfernen oder ein VI als Haupt-VI der Bibliothek ernennen.
- Wenn Sie in LabVIEW 7.x oder Vorgängerversionen einen Fehler-Cluster mit einer Case-Struktur verbinden, wird aus dem True-Case der Fehler-Case und aus dem False-Case der Kein Fehler-Case. Wenn Sie dagegen in LabVIEW 8.0 einen Fehler-Cluster mit einer Case-Struktur verbinden, wird aus dem True-Case der Kein Fehler-Case und aus dem False-Case der Fehler-Case.
- Sie können eine Eigenschaft oder Methode im Blockdiagramm erstellen, indem Sie das Bedien- oder Anzeigeelement auf dem Frontpanel mit der rechten Maustaste anklicken und nach Auswahl von **Erstelle»Eigenschaftsknoten** oder **Erstelle»Methodenknoten** die gewünschte Eigenschaft oder Methode auswählen.
- **(Windows)** Mit dem CAN-Kanal-Bedienelement auf der Palette **CAN Controls** können Sie mit der NI-CAN Channel API arbeiten. Weitere Informationen zu den Elementen in der Palette **CAN Controls** finden Sie im Handbuch zu NI-CAN.

- Wenn auf der Seite **Format und Genauigkeit** im Dialogfeld **Eigenschaften** der **Genauigkeitstyp** auf **Signifikante Stellen** eingestellt ist, wird empfohlen, im Feld **Stellen** für Fließkommazahlen mit einfacher Genauigkeit einen Wert von 1 bis 6 und im Feld **Stellen** für Fließkommazahlen mit doppelter und erweiterter Genauigkeit einen Wert von 1 bis 13 zu verwenden.
- Um den Anfangswert der x-Achse auf 0 statt t_0 zu setzen, aktivieren Sie auf der Seite **Skalierungen** des Dialogfelds **Eigenschaften** eines Diagramms, Intensitätsdiagramms oder eines Intensitätsgraphen die Option **t_0 des Signalverlaufs auf x-Achsen ignorieren**, wenn Sie die x-Achse als Skalierung wählen. Wenn die y-Achse zur Skalierung ausgewählt wurde und der digitale Signalverlauf in Form einzelner Kurven angezeigt werden soll, aktivieren Sie die Option **Digitalen Bus erweitern**.
- Sie können mehrere Dateien zeitgleich öffnen. (**Windows**) Wählen Sie zur Anzeige des Standarddialogfelds **Datei»Öffnen** und drücken Sie zur Auswahl mehrerer Dateien die Tasten <Shift> und <Strg>. (**Mac OS**) Drücken Sie die Taste <Shift> oder <Option>. (**Linux**) Drücken Sie die Taste <Shift> oder <Alt>.
- Die Option **Monitor** im Dialogfeld **Web-Dokumentationswerkzeug** wird von allen Browsern unterstützt.
- Wenn Sie in LabVIEW 7.x oder Vorgängerversionen **Werkzeuge»Vergleichen»VIs vergleichen** auswählen und auf die Schaltfläche **Auswählen** klicken, wird das Dialogfeld **VI über Namen auswählen** angezeigt. Wenn Sie in LabVIEW 8.0 **Werkzeuge»Vergleichen»VIs vergleichen** auswählen und auf die Schaltfläche **Auswählen** klicken, wird das Dialogfeld **Ein VI auswählen** angezeigt.
- Mit der Option **Werkzeuge»Vergleichen»VI-Hierarchien vergleichen** können Sie in LabVIEW 8.0 ein VI aus dem Dateisystem oder der Liste mit VIs im Speicher auswählen.
- Um in LabVIEW 7.x zu ähnlichen Objekten einer Unterpalette zu springen, klicken Sie ein beliebiges Blockdiagramm-Objekt mit der rechten Maustaste an und wählen Sie **Palette** aus dem Kontextmenü (... steht dabei für den Namen der Unterpalette, aus der das Blockdiagramm-Objekt stammt). In LabVIEW 8.0 können Sie auch mit der rechten Maustaste auf ein Blockdiagramm-Objekt klicken und dieses durch Auswahl von **Ersetzen».... Palette** durch ein ähnliches Objekt ersetzen. Sie können auch eine Verbindung mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **Einfügen».... Palette** wählen, um ein Objekt aus der am häufigsten genutzten Palette einzufügen. Mit dieser Option wird das ausgewählte Objekt zwischen den beiden verbundenen Objekten eingefügt.

- Wenn Sie in LabVIEW 7.x und Vorgängerversionen den Datentyp einer Typdefinition ändern und die Instanz der Typdefinition aktualisieren, erhalten die Instanzen nur die aktuellen Werte, Voreinstellungen und Privatdaten sowie Untertitel und Beschriftung. In LabVIEW 8.0 können die Instanzen der Typdefinitionen zusätzliche Attribute haben.
- Wenn Sie in LabVIEW 8.0 eine Typdefinition durch ein anderes Bedienelement ersetzen, erhält das Element möglicherweise weniger Attribute der Typdefinition als in LabVIEW 7.x und Vorgängerversionen.
- In LabVIEW werden so oft wie möglich Typdefinitionen verwendet.
- Das Dialogfeld **Breite und Höhe festlegen** wurde in **Objektgröße verändern** umbenannt. Im Listenfeld **Gegenwärtige Größe** dieses Dialogfelds wird neben den Werten für Objekte, deren Größe selbst bei einer anderen **Breite** oder **Höhe** nicht verändert werden kann, ein Sternchen (*) angezeigt.
- Wenn Sie in LabVIEW 7.x ein VI nicht finden, müssen Sie das fehlende VI durch ein VI mit dem gleichen Namen ersetzen. In LabVIEW 8.0 dagegen können Sie ein fehlendes VI durch ein VI mit beliebigem Namen ersetzen. LabVIEW ersetzt alle Exemplare des fehlenden VIs durch das von Ihnen ausgewählte VI.
- In LabVIEW 7.x und Vorgängerversionen können Sie die Ausrichtung von Objekten auf dem Frontpanel oder im Blockdiagramm durch Drücken der Tasten <Strg-A> wiederholen. In LabVIEW 8.0 muss zur Wiederholung einer Ausrichtung von Objekten die Tastenkombination <Strg-Shift-A> gedrückt werden. Mit der Tastenkombination <Strg-A> werden alle Objekte auf dem Frontpanel oder dem Blockdiagramm markiert.
- Sie können festlegen, wann SubVIs geladen werden sollen. Klicken Sie mit der rechten Maustaste auf ein SubVI und wählen Sie aus dem Kontextmenü die Option **Einstellungen** aus. Daraufhin wird das Dialogfeld **Konfiguration für VI-Aufruf** angezeigt. Dieses Kontextmenü gibt es jedoch nur bei VIs, die in einer Anwendung geöffnet sind, bei der VI-Server-Aufrufe möglich sind. Bei Funktionen, Express-VIs, polymorphen VIs oder VIs, die in einer Anwendung geöffnet sind, bei der keine VI-Server-Aufrufe möglich sind (wie Anwendungen auf einem PDA oder FPGA) gibt es das Menü nicht. Bei einem großen VI, das ein anderes aufruft, können Sie die Ladezeit verringern und Speicherplatz sparen, indem Sie im Dialogfeld die Option **Beim ersten Aufruf laden und halten** auswählen. Bei Auswahl dieser Option wird das SubVI erst dann geladen, wenn es vom übergeordneten VI benötigt wird, und nach dem Ende der Operation wird es aus dem Speicher entfernt.

- Wenn Sie in LabVIEW 8.0 Frontpanel-Objekte vom Blockdiagramm aus kopieren und in das Blockdiagramm eines neuen VIs einfügen, erscheint das kopierte Element an einer anderen Stelle als das Original. Die Frontpanel-Objekte werden in der linken oberen Ecke eingefügt, um Überlappungen bzw. das Verschieben in nicht sichtbare Bildschirmbereiche zu vermeiden.
- Der LabVIEW-Anzeigebereich für Jahresangaben wurde von 1904 bis 2038 auf 1600 bis 3000 erweitert.
- Wenn in der **Kontexthilfe** ein Wagenrücklauf angezeigt werden soll, müssen Sie Absätze durch zwei Wagenrückläufe trennen.
- Wenn ein Tastendruck einer Tastenkombination im VI-Menü entspricht, zum Beispiel <Strg-C> oder <Strg-V>, wird kein Ereignis des Typs “Taste gedrückt” ausgelöst, selbst wenn der Menüpunkt aktiviert ist.
- Mit der Eigenschaft “Applikation:Sprache” kann jetzt auch **ko** für Koreanisch ausgegeben werden.
- Mit Hilfe der Methoden “VI-Strings:Exportieren” und “VI-Strings:Importieren” können mehrere VIs in eine Textdatei mit Tags exportiert oder aus einer solchen importiert werden.

National Instruments, NI, ni.com und LabVIEW sind Marken der Firma National Instruments Corporation. Nähere Informationen zu den Marken von National Instruments finden Sie im Abschnitt *Terms of Use* unter ni.com/legal. Sonstige hierin erwähnte Produkt- und Firmenbezeichnungen sind Marken oder Handelsnamen der jeweiligen Unternehmen. Nähere Informationen über Patente auf Produkte von National Instruments finden Sie unter **Hilfe»Patente** in Ihrer Software, in der Datei `patents.txt` auf Ihrer CD oder unter ni.com/patents. Für Urheberrechtshinweise, Nutzungsbedingungen und rechtliche Hinweise betreffend Komponenten, die in USI genutzt werden (Xerces C++, ICU und HDF5), siehe `USICopyrights.chm`.